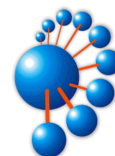




UNIVERSIDAD DE CONCEPCIÓN  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA  
Y CIENCIAS DE LA COMPUTACIÓN



# **CRIPTOGRAMA**

Desarrollo de un software de escritorio para implementar y automatizar procesos en la lógica del juego

**POR**

**Joaquín Ignacio Vásquez Jara**

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción para optar al título profesional de Ingeniero Civil Informático

Patrocinante

Marcela Varas Contreras

Comisión Evaluadora

Lilian Salinas Ayala, Pierluigi Cerulo

Concepción, 27 de noviembre de 2023

© 2023. Joaquín Ignacio Vásquez Jara

Ninguna parte de esta tesis puede reproducirse o transmitirse bajo ninguna forma o por ningún medio o procedimiento, sin permiso por escrito del autor.

Al *@jokeez* de 2013 que comenzó este camino  
y que nunca pensó en rendirse a pesar de  
lo difícil que fue en un principio.

## Agradecimientos

Durante mi etapa universitaria, me enfrenté a uno de los mayores desafíos de mi vida. Este logro no habría sido posible sin el apoyo fundamental de ciertas personas que considero indispensables en este camino llamado universidad. A lo largo de estos últimos años de mi carrera universitaria, si bien he acumulado valiosos conocimientos, he llegado a valorar aún más las relaciones de amistad cultivadas con personas tanto de mi propia carrera como de otras. Muchas de estas amistades se gestaron hace varios años y persisten hasta hoy, algunas tomaron rumbos inesperados a mitad del camino, otros se sumaron más adelante. Siento que sin el respaldo de ellos, alcanzar lo que he logrado no habría sido posible.

Quiero agradecer a quienes estuvieron conmigo desde los primeros días de clases, uno de ellos es Sergio Cifuentes (Checho) que siempre está dispuesto a ayudar, a estudiar hasta tarde o trasnochar si era necesario, tenemos muchas anécdotas juntos dentro de la universidad, también a los chicos de Ingeniería Civil que estuvimos juntos los primeros 2 años. Luego llegó la pandemia, aparecieron nuevas personas que se convirtieron en amigos al poco tiempo, ellos son Nicolás Rojas (Zso) y Marcelo Vargas (Chelo), con quienes estudié y trabajé hasta altas horas de la madrugada, en algún momento nos disfrazamos de personajes de Minecraft en la universidad. En muy poco tiempo se convirtió en una persona importante para mí, Lucas Kraemer que siempre está cuando lo necesito, es el que más me incentiva a los videojuegos. Por último, a quienes fueron mis alumnos en su primer año en mi etapa de monitor y quienes son mis amigos actualmente, Franco, Darling (Jajitas), Catalina (Morri), Álvaro (Adre), Vicente (Schultz), Cristián (Mankeke).

Quiero agradecer a mi familia, en especial a mis padres que me han apoyado en todas mis decisiones y siempre han estado conmigo, a mis hermanas, sobrinas. Agradezco también a algunos docentes del departamento de la carrera quienes me aconsejaron durante el transcurso de esta memoria de título, a mi profesora guía Marcela Varas quien vio un potencial en el prototipo de juego que tenía en un principio, a Julio Godoy quien conoció la parte más rudimentaria del juego y aconsejó a cómo comenzar a escribir este informe, a José Fuentes quien voluntariamente me escuchó cuando ya no sabía como seguir con el rumbo de esta memoria y algunas decisiones importantes que hice fue gracias a él.

A todos ellos, muchas gracias.

## Resumen

Desde hace más de 85 años la televisión se ha encargado de llevar entretenimiento a los hogares mediante programas de concursos del área de entretenimiento, obteniendo algunos grandes audiencias en *rating* y consolidados durante varios años de emisión, además de adaptaciones del formato a nivel mundial. En esta memoria de título se llevará a cabo el desarrollo de software del juego Criptograma, una versión original propia que se establece en sus bases como un formato de programa de concursos para la televisión, que además tiene su propio buscador de contenidos para el desarrollo del juego. El objetivo principal es tener un software único, que aporte valor al producto final y entretenimiento al público, además de llevar a cabo el proceso completo de creación de un software acorde a los conocimientos que debe tener un Ingeniero Civil Informático.

# Índice general

Agradecimientos	i
Resumen	ii
Índice de figuras	vi
<b>1. Introducción</b>	<b>1</b>
1.1. Objetivo general . . . . .	2
1.2. Objetivos específicos . . . . .	2
1.3. Metodología de trabajo . . . . .	2
1.4. Planificación . . . . .	3
<b>2. Marco Teórico</b>	<b>4</b>
2.1. El origen de los juegos de mesa y su transformación a la televisión . . . . .	4
<b>3. Descripción del problema a abordar</b>	<b>8</b>
3.1. Buscador de anagramas . . . . .	11
3.2. Solución propuesta . . . . .	11
<b>4. Análisis de requerimientos</b>	<b>12</b>
4.1. Requerimientos del sistema . . . . .	12
4.2. Requerimientos de arquitectura . . . . .	14
4.3. Casos de Uso . . . . .	14
4.4. Arquitectura de software . . . . .	20
4.5. Modelo de datos . . . . .	21
<b>5. Método de desarrollo</b>	<b>22</b>
5.1. Herramientas utilizadas . . . . .	22
5.2. Creación del software . . . . .	25
5.3. Visualización del software . . . . .	26

---

5.4. Limitaciones del software . . . . .	32
5.5. Evaluación de algoritmos . . . . .	32
<b>6. Testing</b>	<b>35</b>
6.1. Fase Alfa: Quality Assurance . . . . .	35
6.2. Fase Beta: Validaciones de usuarios . . . . .	39
6.3. Fase de Producción . . . . .	41
<b>7. Conclusiones</b>	<b>42</b>
7.1. Trabajo futuro . . . . .	43
<b>Referencias</b>	<b>43</b>
<b>Anexos</b>	<b>45</b>
<b>A. Diseño de gráficas</b>	<b>46</b>
<b>B. Entorno gráfico</b>	<b>48</b>

# Índice de tablas

4.1. Caso de uso (UC1) Poder registrarse e iniciar sesión . . . . .	16
4.2. Caso de uso (UC2) Crear palabras . . . . .	16
4.3. Caso de uso (UC3) Crear paneles . . . . .	17
4.4. Caso de uso (UC4) Visualizar paneles . . . . .	17
4.5. Caso de uso (UC5) Crear partida . . . . .	18
4.6. Caso de uso (UC6) Ver reglas del juego . . . . .	18
4.7. Caso de uso (UC7) Modificar datos del usuario . . . . .	19
5.1. Resultados de tiempo de ejecución promedio en milisegundos . . . . .	33



# Índice de figuras

4.1. Diagrama de casos de uso . . . . .	15
4.2. Diagrama de la arquitectura propuesta . . . . .	20
4.3. Modelo Entidad-Relación asociado a los datos del usuario y al contenido del juego. . . . .	21
5.1. Caja de pulsadores vista desde arriba . . . . .	23
5.2. Vista dentro de la caja de pulsadores . . . . .	23
5.3. Caja de pulsadores vista desde el lado izquierdo . . . . .	24
5.4. Vista del inicio de sesión del usuario . . . . .	27
5.5. Vista de la pantalla Crear palabras . . . . .	28
5.6. Vista del menú de la pantalla Crear paneles . . . . .	28
5.7. Vista del creador del panel Criptocódigo . . . . .	29
5.8. Vista de la pantalla Crear partida . . . . .	29
5.9. Vista de la pantalla de juego Criptocódigo . . . . .	30
5.10. Vista de la pantalla Reglas . . . . .	31
5.11. Vista de la pantalla Datos del usuario . . . . .	31
5.12. Resultados de tiempo de ejecución promedio en milisegundos. Gráfico log-log	34
5.13. Resultados de tiempo de ejecución promedio en milisegundos. Gráfico log-log	34
6.1. Vista del juego en una pantalla de 43 pulgadas. . . . .	37
6.2. Vista de la pantalla Creador de paneles . . . . .	38
6.3. Vista de la pantalla Inicio de sesión . . . . .	39

# Capítulo 1

## Introducción

Los programas de concursos que se emiten por televisión existen desde hace muchos años y, a medida que la tecnología ha avanzado con el tiempo, se han digitalizado de tal forma que el televidente también quiere sentirse parte del propio juego estando detrás de la pantalla.

Desde su aparición en televisión, estos juegos se han implementado con sistemas manuales donde el participante debe estar presente en un estudio de televisión para ser parte del juego, también el público presente dentro del programa se hace partícipe de encuestas como el comodín del público en *¿Quién quiere ser millonario?* [1] o las tres personas de confianza del participante en *El Juego de los Anillos* [2]. Actualmente los programas de concursos para estar a la vanguardia han estado usando desde los años 2000 sistemas tecnológicos haciendo que el transcurso del juego sea más rápido y eficiente, ya que estos tienen un tiempo de emisión en pantalla que no se debe exceder, además de ser menos tolerantes a fallos humanos por su automatización, esto no quiere decir que los sistemas sean perfectos.

El problema que se aborda apunta al desarrollo del software del juego *Criptograma*. Este juego es una idea original, todas las reglas y funciones, al estar ya creadas, no son parte de esta memoria, solo se apunta a la implementación del software.

En resumen, *Criptograma* consiste en resolver anagramas y alfagramas en paneles donde se muestran las letras de una palabra y otra palabra muestra espacios por completar con el anagrama o alfagrama a resolver. La cantidad de participantes es de dos a cuatro personas. Hay un tiempo determinado para resolver cada panel, o bien, puede ser resuelto presionando un pulsador. Cada panel tiene un cierto puntaje que se determina por el tipo de panel. Hay cinco tipos de paneles diferentes: *Palabra anagrama*, *Frase alfagrama*, *Contrarreloj*, *Contrarreloj múltiple* y *Criptocódigo*. Este último resuelve un anagrama dentro de una frase

alfagrama, es resuelto por el participante con mayor puntaje de todo el juego en un tiempo determinado.

## 1.1. Objetivo general

Desarrollar un software de escritorio que implemente las reglas de *Criptograma*, se debe visualizar en una pantalla con una interfaz intuitiva, debe mejorar el proceso de creación de paneles. Por otra parte, el software debe hacer uso de una interfaz física.

## 1.2. Objetivos específicos

1. Diseñar la arquitectura del software a crear.
2. Implementar las reglas de *Criptograma* y un entorno gráfico.
3. Implementar un buscador de anagramas.
4. Automatizar el proceso de creación de palabras y frases en los alfagramas.
5. Diseñar una base de datos que almacene la información de los paneles.
6. Diseñar e implementar un mando con pulsadores.

## 1.3. Metodología de trabajo

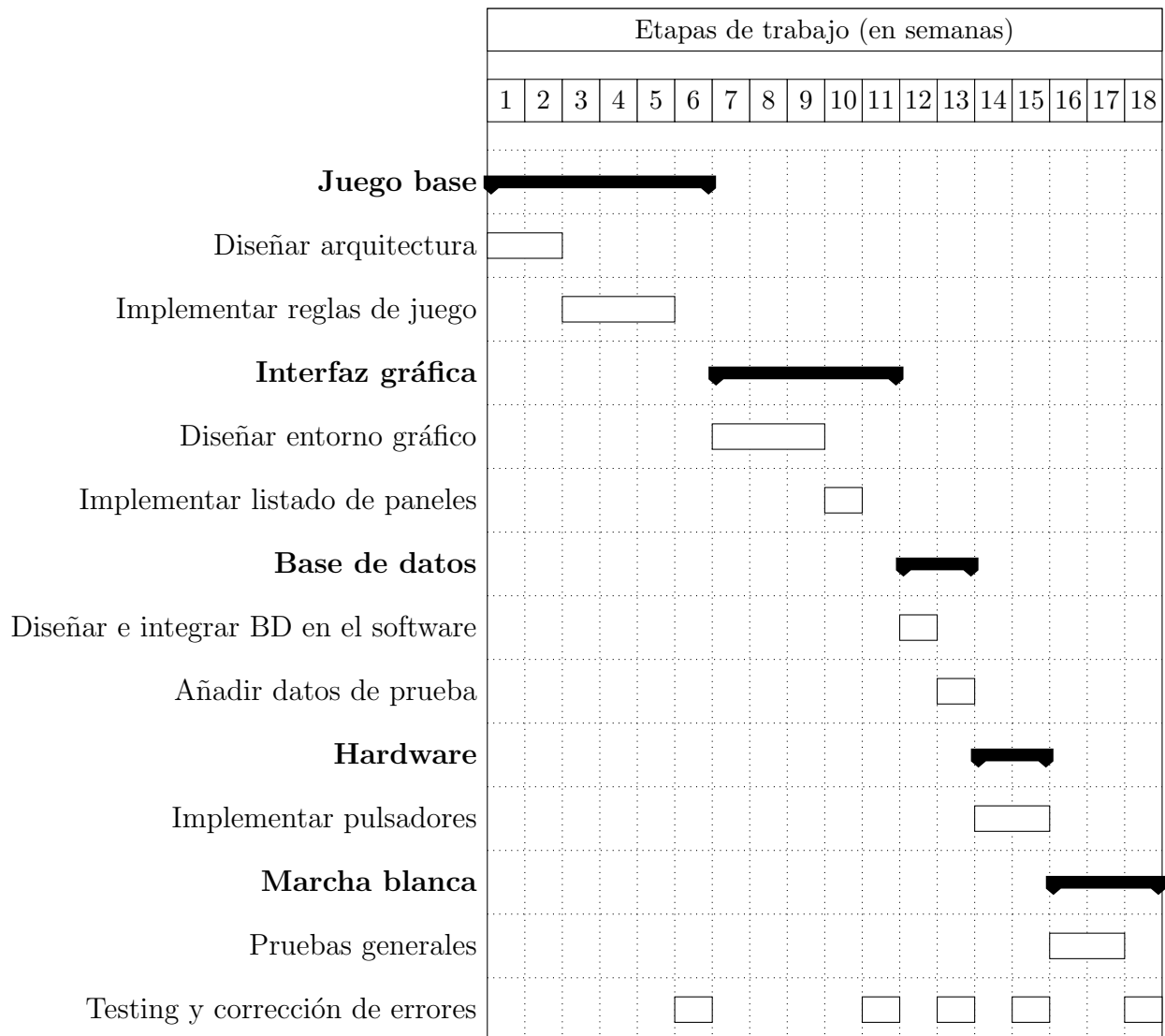
- Cada semana se realiza una reunión con la profesora guía donde se revisa mediante un informe el avance según la planificación propuesta en la sección 1.4.
- La implementación tiene un control de versiones en un repositorio de Git de acceso colaborativo para el alumno y para la profesora guía.
- En las pruebas generales se lleva a cabo el juego en tiempo real con personas en un espacio físico, donde se prueba el correcto funcionamiento del software.

### 1.3.1. Implementación

- Para el desarrollo de las reglas y el entorno gráfico de *Criptograma*, se utiliza el lenguaje de programación Python. Para el diseño de las gráficas se utiliza Adobe Illustrator.

- Se utiliza MySQL para la base de datos. Estos datos son alojados en un servidor externo propio.
- Los pulsadores son cuatro botones físicos tipo arcade controlados por una placa Arduino Leonardo, el desarrollo del código se realiza en el lenguaje Arduino.

## 1.4. Planificación



# Capítulo 2

## Marco Teórico

### 2.1. El origen de los juegos de mesa y su transformación a la televisión

Los juegos de mesa son actividades lúdicas que se juegan sobre una superficie plana, generalmente un tablero, y requieren la participación de uno o más jugadores. Estos juegos involucran reglas específicas que determinan el objetivo y las acciones permitidas durante el juego. Pueden abarcar una amplia variedad de géneros y temáticas, desde juegos de estrategia y cooperación hasta juegos de azar y habilidad.

Las aplicaciones de juegos han revolucionado el mundo de los juegos de mesa al proporcionar nuevas formas de jugar, interactuar y disfrutar de estos juegos. Estas aplicaciones han llevado los juegos de mesa al ámbito digital, permitiendo que los jugadores puedan disfrutar de ellos en dispositivos móviles y en línea. Una de las principales ventajas de estas aplicaciones es la posibilidad de jugar en solitario o conectarse con otros jugadores en línea, lo que amplía la experiencia de juego y permite jugar en cualquier momento y lugar. Además, las aplicaciones ofrecen tutoriales interactivos, ayudando a los jugadores a aprender las reglas y estrategias de los juegos de manera más rápida y sencilla. También brindan la opción de personalizar las partidas, explorar diferentes modos de juego, acceso a expansiones o contenido adicional y actualizaciones. Esto mismo hace que las aplicaciones de juegos sean un aporte a la innovación y más accesibles, revitalizando la popularidad de los juegos de mesa clásicos y atrayendo a nuevos jugadores a este emocionante mundo.

El mundo de la televisión no se ha quedado atrás y desde sus inicios ha querido lle-

var al público una adaptación de algunos juegos de mesa más allá de lo que estos pueden ofrecer e incentivando a los televidentes a que jueguen desde sus casas tomando el lugar de los concursantes que se encuentran participando, y también un propio formato de juegos (llamado programa de concursos) donde al ganador se le ofrecen premios que pueden ser físicos o en dinero. Estos suelen desarrollarse en un estudio de televisión con grandes paneles de juego acompañados de un modelo, una mesa para los participantes y un presentador quien lleva el ritmo de juego.

El primer programa de concursos que fue transmitido por televisión se realizó en Londres el 31 de mayo de 1938 emitido por la BBC, “Spelling Bee” que es una adaptación del formato de radio adaptado por la misma cadena de TV, fue un juego de deletrear palabras donde el presentador le preguntaba a los participantes que deletrearán una palabra con un concepto definido para que se entendiera a qué palabra se estaban refiriendo y luego les mencionaba la palabra a deletrear. El concursante si erraba a la palabra, se continuaba con el siguiente o hasta que alguno de estos respondiera correctamente y se llevaba los puntos en juego. Este programa de solo quince minutos tuvo cinco emisiones en vivo y logró el Record Guinness por ser el programa más antiguo del mundo [3].

Uno de los programas con mayor duración en la televisión mundial es “Wheel of Fortune” (La Ruleta de la Suerte) [4] debutando en 1975 en Estados Unidos. Creado por Merv Griffin y basado en el clásico juego del ahorcado, en este programa los concursantes deben resolver paneles de acertijos de palabras para ganar dinero y premios haciendo girar una ruleta. Cada juego presenta a tres concursantes o tres equipos de dos concursantes (en ediciones especiales) con un marcador para cada equipo. Cada ronda tiene una categoría y un panel con casilleros en blanco que representan a una letra de la respuesta, además se agrega una pista que ayuda a encontrar la respuesta del panel. Por otra parte, antes de comenzar a responder el panel, cada participante debe girar en su turno de la ruleta, que es un mecanismo de rueda dividido en veinticuatro partes iguales, a estos se les denominan gajos, los cuales poseen un monto en dinero definido y colocados en gran parte de la ruleta, a estos se le agrega un gajo de pierde turno y dos de bancarrota (o quiebra), este último además de quitar el turno pierde el monto acumulado en el marcador del participante. Este formato de concursos ha tenido diversas adaptaciones internacionales en más de 50 países, algunos todavía en emisión.

A pesar de que estos programas de concursos mencionados anteriormente eran bastante sencillos y su mecanismo era manual, la tecnología se ha encargado de automatizar ciertos procesos de juego con el uso de software para que estos puedan desarrollarse más rápido y con menos tolerancia a fallos, además de agilizar el tiempo de preparación y grabación del propio programa. Un ejemplo clásico es “Who Wants to Be a Millionaire” (¿Quién quiere ser millonario?) donde cada participante se enfrenta a quince preguntas cada

una con 4 alternativas donde la dificultad crece a medida que van avanzando en las preguntas, estas tienen un monto en dinero determinado que aumenta a medida que acumule respuestas correctas. Notar que este programa de concursos tiene varios comodines disponibles para el concursante que puede usar en el momento que estime conveniente. El comodín del público somete a votación al público presente en el estudio mediante un control remoto que tiene cuatro botones con las letras de cada alternativa donde solo puede presionar una letra y confirmar su elección con el botón del mismo nombre, para esto solo tienen diez segundos y el resultado de la votación se muestra en un gráfico de barras con los porcentajes de cada alternativa. Tomando en cuenta que este proceso tomaría bastante tiempo si fuese manual, el uso de software hace que en solo diez segundos el participante pueda tomar una decisión sobre su respuesta definitiva.

Otro de los programas de concursos que llevan más allá el uso de la tecnología es “5 Golden Rings” (El Juego de los Anillos) [5] donde tres equipos de dos participantes interactúan en su turno con cinco anillos de distinto tamaño en una pantalla de ocho metros donde deben responder a una pregunta en treinta segundos, para esto deben ubicar un anillo en una posición estratégica para coincidir con la respuesta que está oculta, si la respuesta se encuentra dentro del anillo se considera correcta, de lo contrario, el siguiente equipo responde con el anillo inmediatamente más pequeño. El público también interactúa mediante una aplicación móvil que simula la gran pantalla haciéndose participe del programa, esto no influye en las respuestas ya que los únicos que pueden ayudar al concursante son tres personas de confianza en la ronda múltiple que están ubicados en el público del estudio, estos utilizan la misma aplicación con la diferencia que su respuesta si se ve reflejada antes de que se ubique el anillo físico encima de la pantalla. En este programa en particular podemos ver que se utiliza tanto software como hardware para que el juego se desarrolle correctamente y de manera fluida; que es lo que necesitan estos programas de concursos de televisión, donde su emisión está restringida a un tiempo determinado.

Hace ya casi 85 años del estreno del primer programa de concursos en televisión, estos han ido evolucionando a medida que la tecnología avanza, adaptándose a lo que existe para llevar entretenimiento al público e inclusive modernizando los programas ya existentes como ocurre en La Ruleta de la Suerte, donde en un principio las letras del panel se daban vuelta manualmente, actualmente es una pantalla que las muestra automáticamente.

Hay juegos que se han reinventado a lo largo del tiempo, como el programa de concursos Pasapalabra (The Alphabet Game) donde dos equipos de dos famosos y un participante juegan diversos paneles de juegos para acumular segundos que se utilizan en el juego final donde tiene 85 segundos adicionales para contestar veinticinco preguntas con respuestas que

pueden comenzar o contener las letras de la A a la Z. Los dos participantes deben pasar por este juego final, el ganador es quien logre contestar correctamente todas las preguntas o quien haya respondido la mayor cantidad, en este último caso pasará a jugar en el siguiente programa y así sucesivamente hasta que el dinero acumulado en juego se lo lleve algún participante. Para acumular estos segundos, deberán resolver diversos juegos con sus equipos como El Ahorcado, Sopa de Letras, Memorice, entre muchos otros. Resaltando la mecánica de algunos juegos clásicos que están incorporados en este programa: Memoria, los equipos deben poner en secuencia 9 palabras en 75 segundos; Palabras Cruzadas, los equipos deben adivinar una palabra que está dividida en dos columnas desordenadas con un concepto que fue elegido al comienzo; Palabra Vertical, donde los equipos deben completar un crucigrama; todos estos tienen un concepto que se elige al principio del juego.

Este último programa de concursos trata de acercar al público a participar del programa y que juegue mientras es emitido en televisión, eso hace que sea uno de los programas de entretenimiento más vistos en el último tiempo tanto en Chile como en otros países, como en España donde es líder en sintonía desde su estreno en el año 2000.

Como podemos notar, estos juegos son bastante simples, utilizan conceptos básicos y de cultura general para ser realizados. Cuando llevamos esto a la creación de un software que realice este tipo de juegos, nos damos cuenta de la simpleza o de la complejidad que puede llegar a tener un juego como este. Es el caso de *Criptograma* que es el software del juego a realizar en esta memoria.



# Capítulo 3

## Descripción del problema a abordar

**Criptograma** se define como un juego de entretenimiento de formato adaptable tanto para un programa de concursos de televisión como para un juego de mesa. Inspirado en la utilización de métodos de criptografía en informática y el uso de anagramas, este realiza una combinación de ambas, trayendo un juego de dificultad media al público en general.

Un **anagrama** es el cambio en el orden de las letras de una palabra o frase que da lugar a otra palabra o frase distinta. Por otra parte, un **alfagrama** es un grupo de letras que se forma al reordenar alfabéticamente las letras de una palabra original. Se entiende por **criptograma** como un fragmento de mensaje cifrado y cuyo significado es ininteligible hasta que es descifrado. El contenido del mensaje se modifica siguiendo un determinado patrón, de manera que solo es posible comprender el significado original tras conocer o descubrir el patrón seguido en el cifrado. Para este juego, el patrón a seguir es la resolución de anagramas y alfagramas, esto según el tipo de panel que se está resolviendo.

Este juego consiste en resolver anagramas y alfagramas. Se debe jugar con al menos dos participantes y un máximo de cuatro, y un moderador quien debe llevar la presentación de los paneles y la continuidad del juego. Tiene un tiempo estimado de treinta minutos y un mínimo de cinco paneles por resolver. Los paneles de este juego son los siguientes:

- Palabra anagrama
- Frase alfagrama
- Contrarreloj
- Contrarreloj múltiple
- Criptocódigo

---

Antes de comenzar el panel *Frase Alfagrama*, se debe jugar el panel *Palabra Anagrama* para conocer al participante que comienza a resolver el panel siguiente. Siempre se debe terminar el juego con el panel *Criptocódigo*. Estos paneles pueden repetirse tantas veces como sea necesario a excepción de este último. A continuación, se describe detalladamente cada panel por separado.

## Paneles de juego

### Palabra anagrama

En este panel, se presenta una palabra, y el desafío consiste en descifrar el anagrama oculto en la parte inferior de la misma. Cada dos segundos, se revela al azar una letra del anagrama. El panel termina cuando un participante, mediante el uso del pulsador, logra acertar la palabra correctamente. Por esta razón, este panel no proporciona pistas adicionales. El pulsador se utiliza para solicitar el turno y responder, y no hay un límite de tiempo establecido. En caso de una respuesta correcta, se otorga un punto y el panel se da por finalizado. En caso contrario, se libera el turno, y el juego continúa hasta que algún participante logre adivinar la palabra de manera precisa. En el peor caso, la palabra oculta se revela por completo, y solo se requiere una respuesta acertada para obtener el puntaje.

### Frase alfagrama

En este panel, disponemos de cinco minutos para descifrar una frase presentada en forma de alfagrama y en su versión original con los espacios vacíos. El alfagrama muestra todas las letras necesarias para completar la frase oculta, acompañado de una pista que ayuda al participante a comprender el significado de dicha frase. El participante que ganó el panel *Palabra Anagrama* inicia este panel. El juego se desarrolla por turnos cíclicos, lo que significa que si un participante pierde su turno, su contrincante a la derecha continúa jugando el panel. Se otorga un punto por cada letra acertada, y el participante que resuelve el panel completo correctamente gana la cantidad de puntos equivalente a las letras disponibles hasta ese momento multiplicado por dos. Cada intento fallido de posicionar una letra de forma correcta resulta en una penalización de dos segundos.

## **Contrarreloj**

En este panel, se dispone de dos minutos para resolver un máximo de quince anagramas con la ayuda del pulsador. Cada anagrama está acompañado de una pista que facilita la comprensión del término oculto. El participante debe presionar el pulsador y dar una respuesta correcta para ganar cinco puntos por cada anagrama resuelto. En caso de respuestas incorrectas, se libera el turno y el juego continúa. Se cambia al siguiente anagrama después de intentar responder más de tres veces. El panel puede concluir de dos maneras: al agotarse el tiempo establecido o cuando se llegue al último anagrama del panel.

## **Contrarreloj múltiple**

En este panel, disponemos de dos minutos para resolver las distintas formas en que un anagrama puede ser resuelto. Se presenta una palabra y sus anagramas válidos con espacios vacíos. Este panel no tiene pista, ya que cada anagrama puede tener significados completamente diferentes entre sí. El participante debe presionar el pulsador para responder uno de los anagramas ocultos, y cada respuesta correcta se premia con cinco puntos. El panel concluye cuando todas las palabras ocultas han sido resueltas correctamente.

## **Criptocódigo**

En este panel, que tiene una duración de diez minutos, se debe resolver un anagrama oculto dentro de una frase presentada en forma de alfagrama. El panel incluye una frase escrita en formato de alfagrama, la misma frase oculta con espacios vacíos, y un anagrama con signos de interrogación que indica la cantidad de letras en la palabra oculta. Además, proporciona pistas tanto para la frase en alfagrama como para el anagrama. El anagrama a resolver es una palabra que tiene la misma cantidad de letras que una palabra de la frase en alfagrama. Este panel está designado exclusivamente para el participante que haya acumulado la mayor cantidad de puntos durante todo el juego. El juego concluye cuando dicho participante logra resolver el anagrama oculto. Cada intento fallido de posicionar una letra de manera incorrecta resulta en una penalización de dos segundos por intento.

## 3.1. Buscador de anagramas

Para que podamos buscar anagramas válidos debemos contar con un listado de palabras que sean válidas en el idioma español. Esto puede ser un gran desafío ya que no suelen estar disponibles para el público. Una vez que se dispone de este listado de palabras válidas, el método para encontrar anagramas implica utilizar el anagrama proporcionado y comparar la igualdad tanto en la cantidad de letras como en su disposición con las palabras del listado. Este enfoque facilita la identificación de palabras que comparten las mismas letras que el anagrama, permitiendo así encontrar un listado con los nuevos anagramas.

## 3.2. Solución propuesta

Como ya podremos notar tanto en la complejidad del juego como en la búsqueda de anagramas, es necesario resolver los anagramas y alfagramas en un tiempo determinado, debemos pensar en la creación completa de un software que resuelva la problemática del juego, para esto definimos lo siguiente:

- Analizar los requerimientos que necesita el juego para ser implementado.
- Implementación de un algoritmo de búsqueda de anagramas.
- Diseño de la arquitectura de software.
- Implementación de las reglas del juego.
- Diseño de las gráficas.
- Creación del entorno gráfico utilizando las gráficas del punto anterior.
- Creación de una base de datos en línea para el contenido del juego.
- Implementación de una interfaz física con pulsadores.
- Pruebas y revisión del software con la información de la base de datos.

# Capítulo 4

## Análisis de requerimientos

### 4.1. Requerimientos del sistema

El software a desarrollar en esta memoria de título tiene por objetivo llevar entretención a las personas con un juego que combina algoritmos sencillos como son los anagramas y alfagramas, esto debe realizarse en un tiempo de juego determinado ya que está diseñado para adaptarse a un formato de programa de concursos y los tiempos son acotados. Para esto debemos tener en cuenta varios factores a considerar.

Debemos considerar el tiempo de juego, ya que los programas suelen tener una duración mínima de 22 minutos de emisión para un bloque de 30 minutos, 45 minutos de emisión para un bloque de una hora, esto considera la diferencia de tiempo como espacio publicitario. Es importante que los tiempos de juego se respeten, por lo mismo, los cinco paneles que están definidos en el juego tienen la posibilidad de repetirse para completar el tiempo de emisión y esto depende del tiempo disponible para que estos puedan ser jugados correctamente, el único que no se puede repetir es el panel *Criptocódigo* por ser el último panel del juego.

Tomando en cuenta lo anterior, necesitamos que el tiempo de los paneles se puedan distribuir de dos formas distintas: por tiempo determinado o por pulsadores, esto quiere decir que los paneles como *Palabra anagrama* a pesar de tener un tiempo determinado, puede acortarse aún más, haciendo que el participante que adivine correctamente detenga el tiempo y se pueda dar término anticipado del panel. Además, tenemos los paneles *Contrarreloj* y *Contrarreloj múltiple* que evidentemente tienen un tiempo determinado para el desarrollo del panel, independiente que las reglas del juego utilicen además los pulsadores para adivinar el anagrama a resolver dentro del mismo panel. En resumen, la cantidad de paneles puede

ser modificada por el usuario, para que estos puedan acomodarse a los tiempos del juego completo. Además de incorporar pulsadores.

Mostrar la información en pantalla es tan importante como el tiempo de juego, por lo que al adivinar las letras, palabras o frases, estas deben ser descubiertas en el menor tiempo posible haciendo de esto un panel más dinámico para el participante. En particular, en el panel *Palabra anagrama* se debe definir un tiempo determinado para cada letra que se muestra. Es decir, la información en pantalla debe mostrarse en el menor tiempo posible apenas sea adivinada.

Esta información debe mostrarse ordenadamente y de tal forma que diferenciamos cuales son los distintos paneles dentro del juego, también debemos preocuparnos de las pantallas que dan inicio y continuidad al proceso de juego. Las pantallas que se necesitan al menos para una buena continuidad son: inicio del programa, reglas del juego, creación de paneles, creación de anagramas y alfagramas y paneles de juego. La visualización de estas pantallas depende de la naturaleza del usuario, las cuales se especifican más adelante por tipo de usuario.

El software debe ser ejecutado en un ordenador de escritorio, este debe adaptarse a la pantalla principal que es el foco de atención de los participantes y debe mostrar información relevante del juego como los anagramas y alfagramas, tiempo transcurrido o restante (según corresponda), pistas (si corresponden) y el puntaje de cada participante.

Este software debe diferenciar los tipos de usuarios que accedan a él, existiendo tres tipos distintos:

- **Administrador:** Usuario que tiene acceso a todo, además de asignar roles a los usuarios mediante una palabra clave.
- **Moderador:** Usuario que está a cargo de llevar a cabo el juego.
- **Editor:** Usuario que provee la información relacionada a la creación de palabras y al panel de juego.

El participante no es considerado como un tipo de usuario, ya que para crear una partida es el moderador quien agrega a los participantes y no es necesario que se registren para que estos puedan jugar.

La información debe ser almacenada en una base de datos en línea para que el juego se lleve a cabo correctamente. Debemos diferenciar las bases de datos en la información relacionada con el juego y la información de los usuarios.

En base a lo mencionado anteriormente, se identificaron los requerimientos asocia-

dos al sistema. Para cada uno de estos, se realizó una clasificación en funcionales y no funcionales.

#### **4.1.1. Requerimientos funcionales**

- R1. Proveer la funcionalidad de registrarse e iniciar sesión a los usuarios.
- R2. Poner a disposición la información relacionada con las reglas del juego.
- R3. Modificar los datos personales de registro del usuario.
- R4. Crear partidas de juego, incluyendo una sección donde el moderador pueda agregar a los participantes y los paneles de juego a utilizar.
- R5. Crear paneles de juego, incluyendo la creación de anagramas, alfagramas, tiempo de juego y pistas (si corresponden).

#### **4.1.2. Requerimientos no funcionales**

- R1. Debe ser intuitivo, de fácil uso y mantenimiento para el usuario.
- R2. Los datos del usuario que se almacenan en la base de datos deben proveer seguridad y confidencialidad en la información.
- R3. Se debe proporcionar escalabilidad para las siguientes versiones del software.

## **4.2. Requerimientos de arquitectura**

La arquitectura del sistema debe soportar la comunicación de información entre el repositorio de datos y la aplicación de escritorio. Para esto último, es deseable que el usuario esté conectado a Internet.

## **4.3. Casos de Uso**

Para identificar de mejor manera las interacciones entre el usuario y el software, se presenta el diagrama de casos de uso para la aplicación de escritorio. En el diagrama de la figura 4.1 se incluye un indicador de los requerimiento funcionales que están relacionados con los casos de uso descritos.

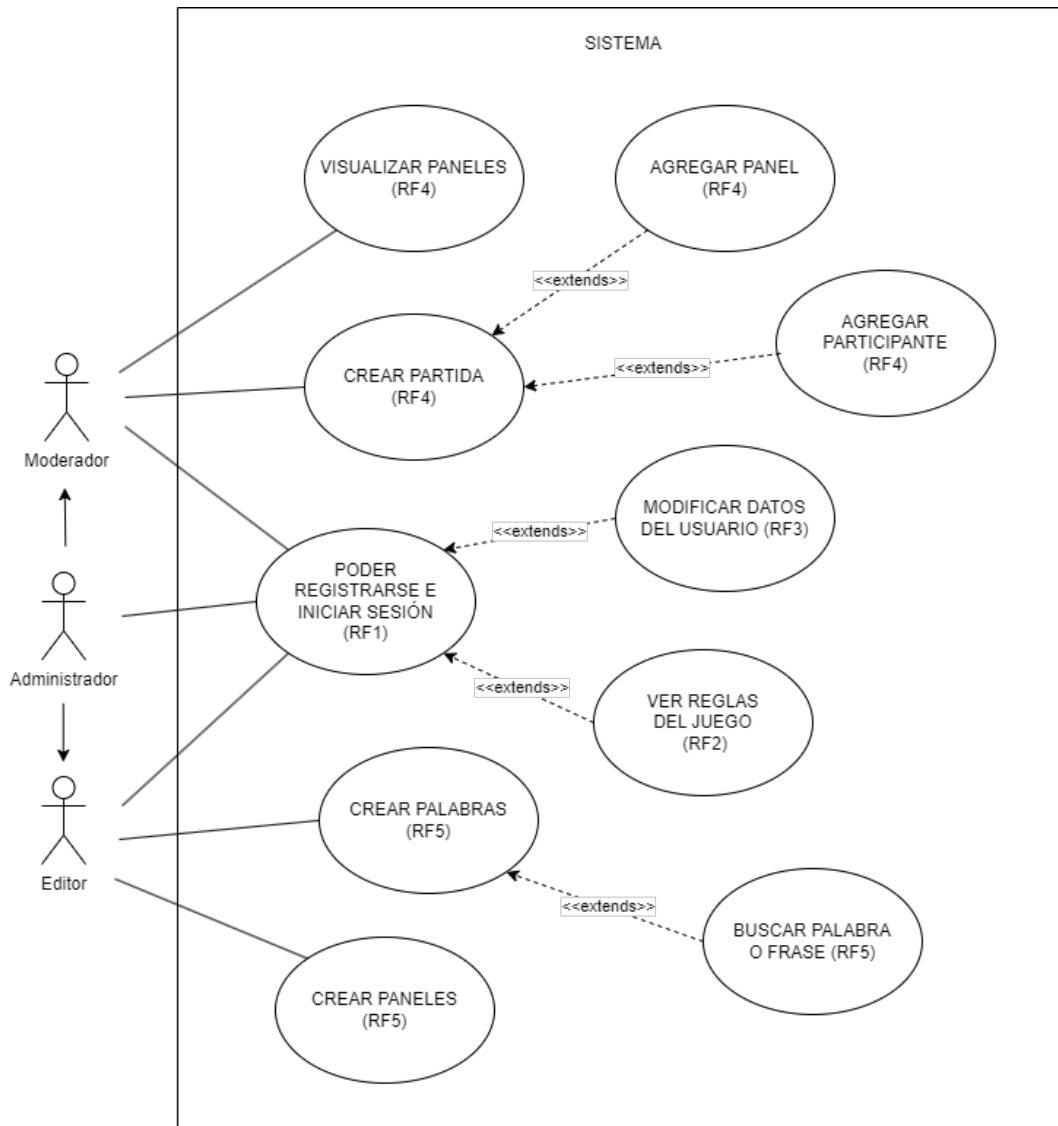


Figura 4.1: Diagrama de casos de uso

A continuación, se documentan los principales casos de uso de la figura 4.1. En todos los casos, el administrador es un actor principal además del mencionado en cada tabla, no se menciona por separado porque el rol que cumple es el mismo en cada caso de uso.



<b>Caso de Uso</b>	(UC1) Poder registrarse e iniciar sesión
<b>Actores principales o usuarios</b>	Administrador, Moderador y Editor
<b>Personal involucrado e intereses</b>	Usuario: Busca iniciar sesión y de no tener cuenta, poder registrarse en la plataforma.
<b>Precondiciones</b>	El Usuario debe iniciar el programa.
<b>Resultado</b>	Se obtiene acceso a la plataforma.
<b>Escenario de éxito</b>	<ol style="list-style-type: none"> <li>1. El usuario debe abrir el programa <i>Criptograma</i>.</li> <li>2. El usuario debe seleccionar el enlace <i>Registrarse</i> para crear una cuenta, para esto debe agregar un nombre de usuario, una clave y guardar los datos.</li> <li>3. El usuario debe iniciar sesión con su nombre de usuario y clave.</li> <li>4. El usuario debe presionar el botón <i>Entrar</i>.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>3'. El usuario no posee cuenta. Ir al paso 2.</li> <li>4'. Problemas de conexión. Reintentar paso 4.</li> <li>4". Usuario y/o clave incorrectos. Volver al paso 3.</li> </ol>

Tabla 4.1: Caso de uso (UC1) Poder registrarse e iniciar sesión

<b>Caso de Uso</b>	(UC2) Crear palabras
<b>Actores principales o usuarios</b>	Editor
<b>Personal involucrado e intereses</b>	Editor: Buscar el alfagrama y anagramas en base a una palabra escrita en el cuadro <i>Palabra o frase</i> .
<b>Precondiciones</b>	El Editor debe ingresar a la sección <i>Crear palabras</i> del menú.
<b>Resultado</b>	Se obtienen los resultados del alfagrama generado y los anagramas encontrados en el leuario.
<b>Escenario de éxito</b>	<ol style="list-style-type: none"> <li>1. El Editor debe ingresar en el cuadro <i>Palabra o frase</i> una palabra de 4 o más letras.</li> <li>2. El Editor debe presionar la tecla <i>Enter</i> o el botón <i>Buscar</i>.</li> <li>3. El programa entrega el alfagrama y los anagramas encontrados.</li> <li>4. Si lo desea, el Editor puede seleccionar dos veces el cuadro <i>Palabra o frase</i>, <i>Alfagrama</i> o <i>Anagramas</i> para copiar su contenido en el portapapeles de su ordenador.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>2'. Error en la búsqueda de anagramas. Se debe ingresar una palabra de 4 a 12 letras.</li> <li>2". Tiempo de búsqueda excesivo. Al usuario se le da un estimado de tiempo antes de buscar la información. Solo tiene que esperar.</li> </ol>

Tabla 4.2: Caso de uso (UC2) Crear palabras

<b>Caso de Uso</b>	(UC3) Crear paneles
<b>Actores principales o usuarios</b>	Editor
<b>Personal involucrado e intereses</b>	Editor: Crear un panel de juego a la base de datos.
<b>Precondiciones</b>	El Editor debe ingresar a la sección <i>Crear paneles</i> del menú.
<b>Resultado</b>	Se guarda en la base de datos la información relacionada con el tipo de panel de juego elegido.
<b>Escenario de éxito</b>	<ol style="list-style-type: none"> <li>1. El Editor elige uno de los 5 tipos de paneles existentes en el juego.</li> <li>2. El Editor completa los campos vacíos de los cuadros respectivos con la información relacionada de la etiqueta de la izquierda y presiona el botón <i>Guardar</i>.</li> <li>3. Si lo desea, el Editor puede buscar con el ID los paneles guardados en la base de datos y presionar la tecla <i>Enter</i> o presionar el botón <i>Buscar</i>.</li> <li>4. Si lo desea, el Editor puede modificar las palabras o frases sobrescribiendo la información mostrada en los campos de texto, y luego presionar el botón <i>Editar</i>.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>2'. Problemas de conexión. Reintentar paso 2.</li> <li>3'. Problemas de conexión. Reintentar paso 3.</li> <li>3". El ID puede ser no válido. Se muestran los campos de texto con un guion.</li> <li>4'. Problemas de conexión. Reintentar paso 4.</li> </ol>

Tabla 4.3: Caso de uso (UC3) Crear paneles

<b>Caso de Uso</b>	(UC4) Visualizar paneles
<b>Actores principales o usuarios</b>	Moderador
<b>Personal involucrado e intereses</b>	Moderador: Crear un panel de juego a la base de datos.
<b>Precondiciones</b>	El Moderador debe ingresar a la sección <i>Visualizar paneles</i> del menú.
<b>Resultado</b>	Se busca en la base de datos la información relacionada con el tipo de panel de juego elegido.
<b>Escenario de éxito</b>	<ol style="list-style-type: none"> <li>1. El Moderador elige uno de los 5 tipos de paneles existentes en el juego.</li> <li>2. El Moderador puede buscar con el ID los paneles guardados en la base de datos y presionar la tecla <i>Enter</i> o presionar el botón <i>Buscar</i>.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>2'. Problemas de conexión. Reintentar paso 2.</li> </ol>

Tabla 4.4: Caso de uso (UC4) Visualizar paneles

<b>Caso de Uso</b>	(UC5) Crear partida
<b>Actores principales o usuarios</b>	Moderador
<b>Personal involucrado e intereses</b>	Moderador: Busca crear un juego completo agregando jugadores y el número de ID a los paneles.
<b>Precondiciones</b>	El Moderador debe ingresar a la sección <i>Crear Partida</i> del menú.
<b>Resultado</b>	Se obtienen los paneles de juego con toda la información asociada a estos y se puede comenzar a jugar.
<b>Escenario de éxito</b>	<ol style="list-style-type: none"> <li>1. El Moderador debe ingresar en cada panel un número correspondiente al ID del panel a agregar.</li> <li>2. El Usuario debe seleccionar la cantidad de jugadores presionando en el recuadro de jugadores la cantidad de veces que sea necesario.</li> <li>3. Si lo desea, el Moderador puede agregar un nombre a cada jugador, esto solo le servirá al Moderador como guía.</li> <li>4. Si lo desea, el Moderador puede reiniciar los valores ingresados con anterioridad a los valores por defecto presionando el botón <i>Reiniciar datos</i>.</li> <li>5. El Usuario debe presionar el botón <i>Jugar</i> del panel respectivo para comenzar el panel de juego.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>1'. El ID puede ser inexistente. Debe verificar la información en la sección <i>Visualizar paneles</i> del caso de uso <i>UC4</i>.</li> <li>5'. Error en la obtención de los datos. No está conectado a Internet.</li> </ol>

Tabla 4.5: Caso de uso (UC5) Crear partida

<b>Caso de Uso</b>	(UC6) Ver reglas del juego
<b>Actores principales o usuarios</b>	Administrador, Moderador y Editor.
<b>Personal involucrado e intereses</b>	Usuario: Busca revisar las reglas del juego de cada panel.
<b>Precondiciones</b>	El Usuario debe ingresar a la sección <i>Reglas</i> del menú.
<b>Resultado</b>	Se obtienen las reglas de cada panel de juego.
<b>Escenario de éxito</b>	1. El Usuario debe seleccionar uno de los paneles de juego y presionar en la etiqueta del nombre del panel.
<b>Excepciones</b>	-

Tabla 4.6: Caso de uso (UC6) Ver reglas del juego

<b>Caso de Uso</b>	(UC7) Modificar datos del usuario
<b>Actores principales o usuarios</b>	Administrador, Moderador y Editor
<b>Personal involucrado e intereses</b>	Usuario: Busca revisar y/o modificar su tipo de usuario o clave.
<b>Precondiciones</b>	El Usuario debe ingresar a la sección <i>Datos del usuario</i> del menú.
<b>Resultado</b>	El usuario realizó alguna modificación a sus datos.
<b>Escenario de éxito</b>	<ol style="list-style-type: none"> <li>1. El usuario puede modificar su tipo de usuario presionando el cuadro con el tipo de usuario actual tantas veces como tipos de usuario existan disponibles. Luego, deberá agregar la palabra clave. Luego, ir a paso 3.</li> <li>2. El usuario puede modificar su clave de acceso a la plataforma. Deberá escribir en los cuadros respectivos una clave nueva y repetir la clave ingresada. Luego, ir a paso 3.</li> <li>3. El Usuario presiona el botón <i>Cambiar</i>.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>1'. El Usuario no posee la palabra clave. Esta clave es proporcionada por el Administrador.</li> <li>3'. Problemas de conexión. Reintentar paso 3.</li> </ol>

Tabla 4.7: Caso de uso (UC7) Modificar datos del usuario

## 4.4. Arquitectura de software

Teniendo en cuenta los elementos que debe tener esta arquitectura, se decide utilizar como paradigma base una arquitectura más simplificada, el modelo C4 que explica los diferentes modelos de abstracción según el nivel de profundidad dependiendo de la complejidad del proyecto a desarrollar. Por su naturaleza, lo más apropiado es utilizar hasta el nivel dos de componentes.

En la Figura 4.2 se presenta el diagrama de la arquitectura propuesta.

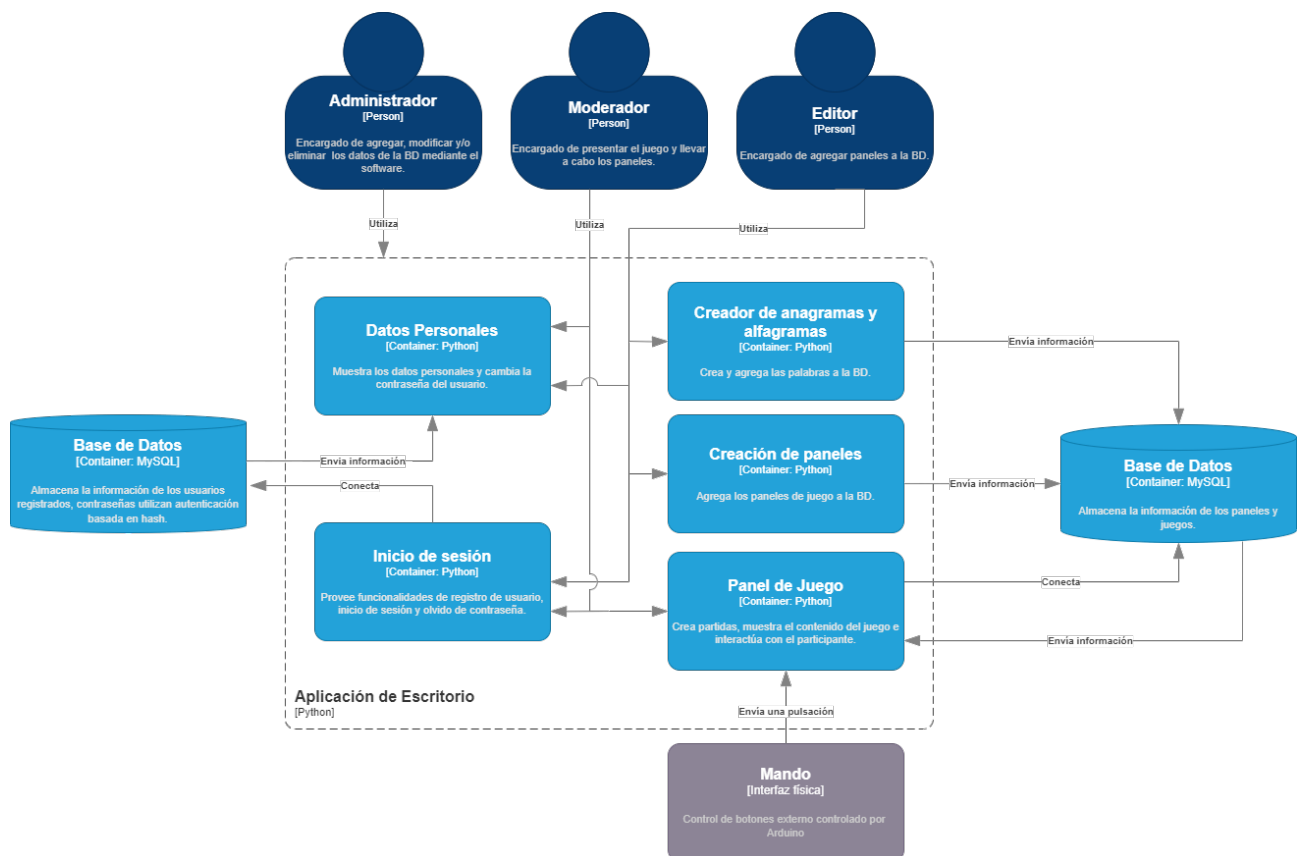


Figura 4.2: Diagrama de la arquitectura propuesta

## 4.5. Modelo de datos

El modelo de datos de este software debe proveer el soporte necesario para almacenar la información asociada a los usuarios como también el contenido de los paneles de juego. También debe permitir representar estos elementos de forma adecuada y organizarlos eficientemente. Si bien el tipo de información que busca almacenar el sistema es propia del contenido del juego, también se busca que el tipo de contenido que se pueda mostrar en un futuro sea ampliable como puede ser un histórico de partidas, un análisis de datos sobre la resolución de los paneles, entre otras ideas más.

En la Figura 4.3 se presenta el modelo Entidad-Relación asociados a los datos del usuario y al contenido del juego.

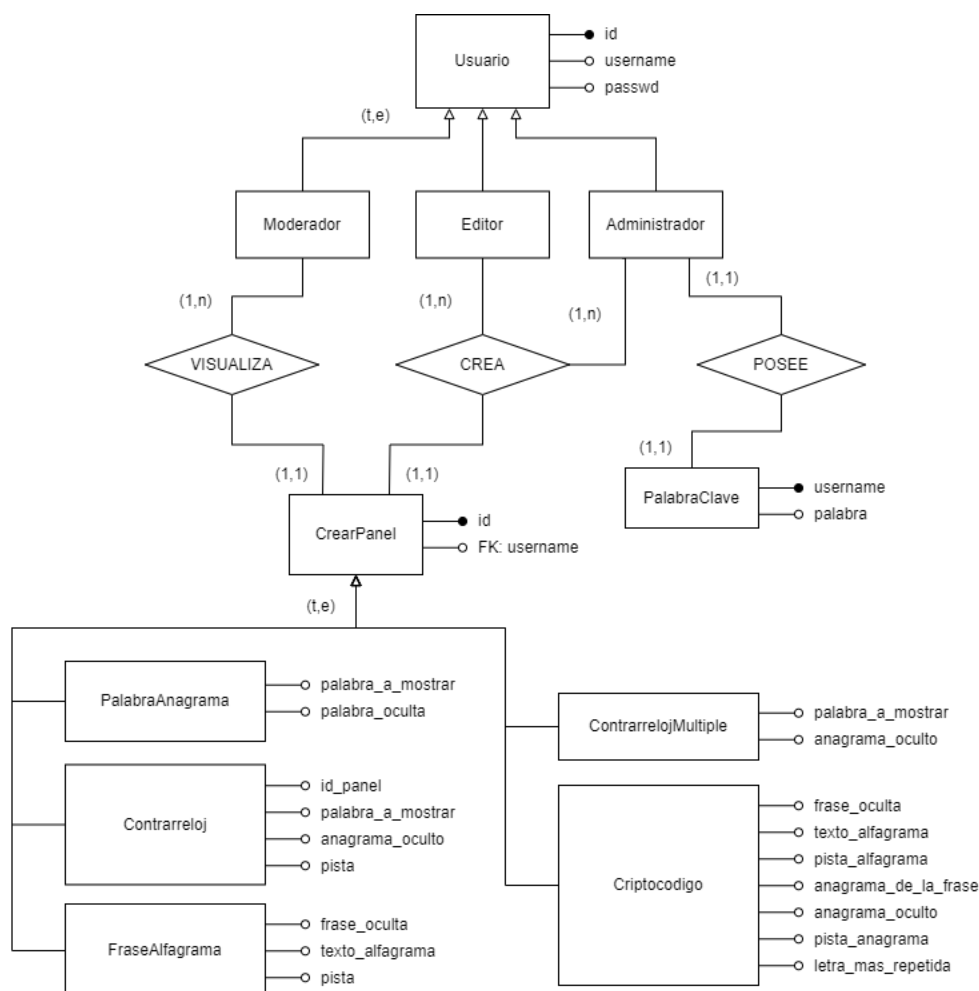


Figura 4.3: Modelo Entidad-Relación asociado a los datos del usuario y al contenido del juego.

# Capítulo 5

## Método de desarrollo

### 5.1. Herramientas utilizadas

#### 5.1.1. Hardware de desarrollo

##### 5.1.1.1. Equipo utilizado

Tanto el trabajo como los resultados de la evaluación se realizaron en un ordenador de escritorio con las siguientes características:

- **Procesador:** Intel Core i5 - 12400 @ 2500 MHz
- **Memoria RAM:** 64 GB DDR4 @ 3200 MHz
- **Disco duro:** SSD M.2 NVMe 1 TB
- **Sistema operativo:** Windows 11 Pro

##### 5.1.1.2. Pulsadores y controlador

Se utilizan 4 botones de tipo arcade que simulan 4 caracteres distintos del teclado y son controlados por el siguiente equipo:

- **Modelo de la placa:** Arduino Leonardo
- **Microcontrolador:** ATmega32u4
- **Memoria Flash:** 32 KB (4 KB usados para el bootloader)

En las figuras 5.1, 5.2 y 5.3 podemos ver la interfaz física y su interior.



Figura 5.1: Caja de pulsadores vista desde arriba

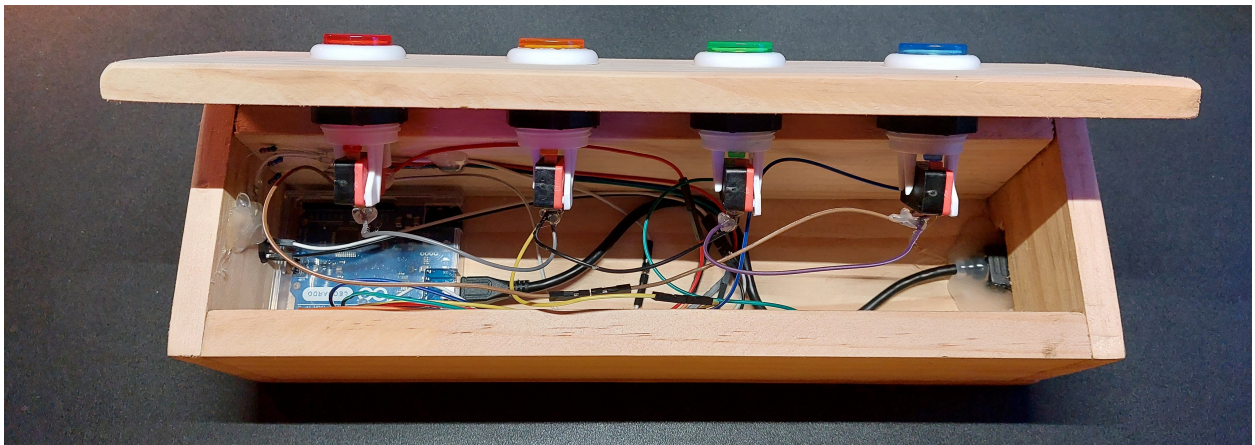


Figura 5.2: Vista dentro de la caja de pulsadores





Figura 5.3: Caja de pulsadores vista desde el lado izquierdo

### 5.1.2. Software de desarrollo

Para un buen desarrollo del software como del controlador de pulsadores, se utiliza:

- **Visual Studio Code** como editor de código y el lenguaje de programación **Python** para el desarrollo del software de juego e implementación de los algoritmos de búsqueda de palabras. Se utiliza la librería *pygame* para el desarrollo del entorno gráfico del programa.
- **MySQL** como el gestor de base de datos, **DBeaver** como aplicación de software cliente.
- **Adobe Illustrator 2023** como herramienta de diseño gráfico para el entorno visual del juego.
- **Arduino IDE** como el entorno de desarrollo y el propio lenguaje de programación Arduino para el desarrollo del control de los pulsadores.

## 5.2. Creación del software

### 5.2.1. Palabras válidas en el español

El diccionario muestra alfabéticamente el significado de palabras en un orden de una forma específica: *lema* que es el nombre de la palabra seguido de su *acepción* que es el significado de la palabra. Si agrupamos los lemas en un listado de palabras, dicho listado se conoce como leuario. El leuario en español que se utiliza en el software tiene 86.128 palabras distintas y su última actualización fue el 17 de octubre de 2010, es una recopilación de palabras hecha por Ismael Olea [8], no es una versión oficial del diccionario español, ya que estos no suelen estar disponibles para el público. Este leuario considera palabras con tres o más letras, incluyendo nombres propios, apellidos y verbos más comunes. Haciendo una revisión más exhaustiva del leuario, notamos que no incluye las conjugaciones de gran parte de las palabras, esto puede significar que varias palabras que son válidas en el español no sean encontradas. Por su naturaleza, el leuario viene ordenado alfabéticamente.

### 5.2.2. Buscador de anagramas

Dado que el leuario es un solo archivo que contiene todas las palabras, para evitar la lectura de un archivo extenso, se procede a dividirlo en varios archivos según la cantidad de letras, organizándolos en orden alfabético.

Para una palabra de largo  $n$ , buscamos en el leuario correspondiente a las palabras de largo  $n$ . Se verifica que las palabras del leuario contengan exactamente las mismas letras, sin importar el orden en el que se encuentren. Como resultado, obtenemos un listado de anagramas.

### 5.2.3. Diseño de gráficos

A pesar de que todas las vistas fueron diseñadas en Adobe Illustrator, tuvieron que replicarse utilizando la herramienta de dibujo de la librería *pygame* para simular tanto el posicionamiento como el dibujado de cada figura y el texto. Esto se debe a que al importar las imágenes directamente, la ventana no es capaz de mostrar los bordes de estas imágenes ya diseñadas. Tanto la paleta de colores como la tipografía están basadas en el videojuego *Neon Horizon* de Marcelo Vargas [9]. En el Anexo A podemos ver algunas vistas del diseño gráfico realizado en Adobe Illustrator.

### 5.2.4. Pulsadores

La figura 5.1 muestra cuatro botones representando los colores rojo, naranja, verde y azul. Estos botones sirven como sustitutos para las teclas Q, W, E y R respectivamente. Cuando un jugador presiona su botón correspondiente, se envía internamente la señal de teclado asociada a esa tecla al software. Esto permite que el software identifique al jugador y le asigne el puntaje correspondiente, si es necesario.

### 5.2.5. Funciones de teclado

Así como la interfaz física vista en el párrafo anterior tiene asignada ciertas teclas, quien controla el software también posee de esta funcionalidad que solo puede ser utilizada cuando se presiona el botón *Jugar* en la pantalla *Crear partida*, estas son:

- *Enter*, para marcar una respuesta como correcta.
- *Backspace*, para marcar una respuesta incorrecta.
- *Espacio*, para comenzar el tiempo de un panel de juego.
- *Shift derecha*, para que el moderador pueda intercambiar su vista con la del jugador.
- *Esc*, para volver a la pantalla *Crear partida*.
- *Izquierda* y *Derecha*, para retroceder y avanzar los paneles en el juego *Contrarreloj*.

Y por razones evidentes, las teclas Q, W, E y R al estar ya asignadas, el moderador también las puede presionar, a pesar de que no le corresponde su utilización.

## 5.3. Visualización del software

En la figura 5.4 podemos ver la pantalla principal donde el usuario debe ingresar su nombre de usuario y una clave para entrar a la plataforma. Podemos ver que en la figura el usuario intenta iniciar con una clave no válida. Esta validación se realiza directamente con la base de datos de usuarios.

La figura 5.5 muestra la sección de creación de palabras y frases. Podemos buscar palabras con cuatro o más letras para que pueda realizar la consulta al buscador de anagramas explicado en la sección 5.2.2, en todos los casos se entrega el alfagrama, independiente de la cantidad de letras. Como información adicional al usuario, se menciona la cantidad de letras de la palabra

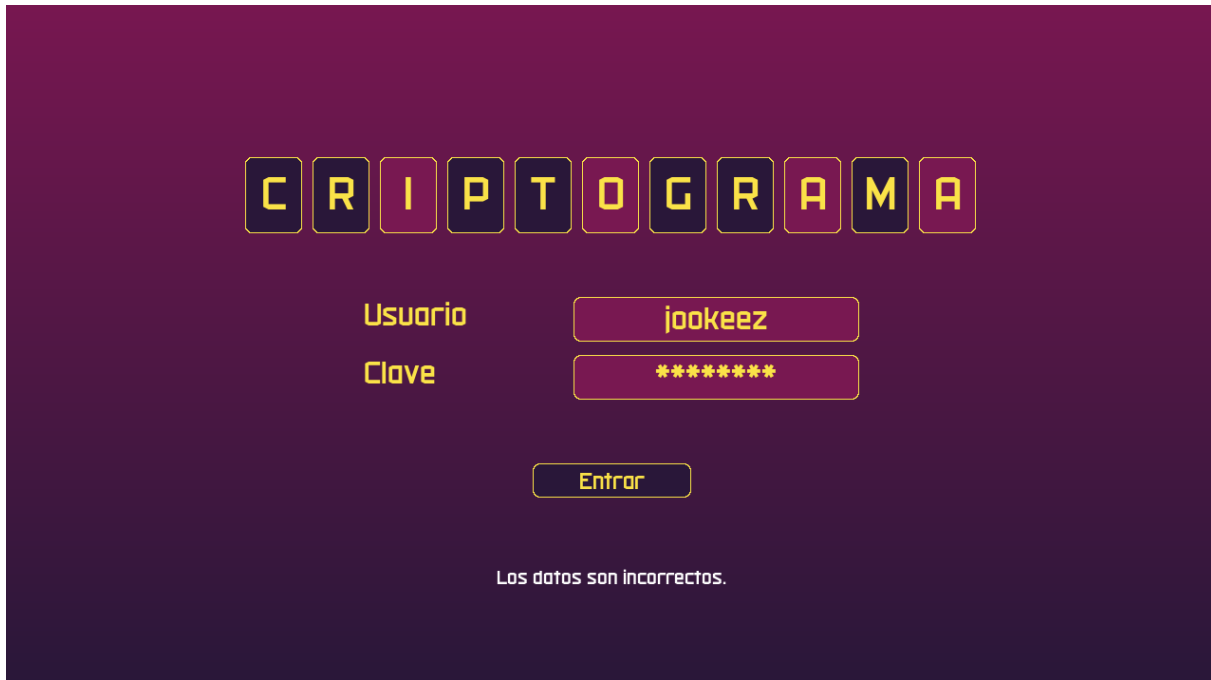


Figura 5.4: Vista del inicio de sesión del usuario

o frase ingresada. En la figura vemos que se busca una palabra de siete letras, los resultados de la búsqueda, y el tiempo real de ejecución de la misma, la cantidad de anagramas encontrados, las palabras anagramas encontradas y el alfagrama. Como información adicional, si el usuario presiona sobre el campo de texto dos veces, la palabra o frase queda en el portapapeles del ordenador en uso. También, al buscar una palabra o frase puede presionar la tecla *Enter* para realizar la búsqueda, tiene la misma funcionalidad que el botón *Buscar*.

La figura 5.6 muestra los paneles que podemos crear en la base de datos. Al seleccionar uno de ellos, podemos ver una nueva vista con el detalle que necesitamos para agregar en el panel de juego. En la figura 5.7 vemos que con buscar el ID número dos, obtenemos la información relacionada al panel *Criptocódigo*. Al igual que en la vista anterior, puede presionar la tecla *Enter* para realizar la búsqueda.

La figura 5.8 nos muestra el resumen de la partida de juego con los paneles que se pueden jugar dentro de cuarenta y cinco minutos. Esto siempre queda a criterio del tiempo real, para esto se presenta el tiempo jugado en esta pantalla. Notamos que esta pantalla cuenta con siete paneles, varios ID predefinidos que cuentan con información en la base de datos, estos pueden cambiarse modificando el número y asegurando que el panel exista, también vemos los nombres y colores de los participantes y su puntaje respectivo total en todo el juego. Al presionar el número de jugadores, podemos editar dicha cantidad tantas veces como las reglas del juego lo permitan. En la parte inferior, vemos el botón *Reiniciar datos* que vuelve a los



Figura 5.5: Vista de la pantalla Crear palabras



Figura 5.6: Vista del menú de la pantalla Crear paneles



Figura 5.7: Vista del creador del panel Criptocódigo



Figura 5.8: Vista de la pantalla Crear partida

valores por defecto de esta vista.

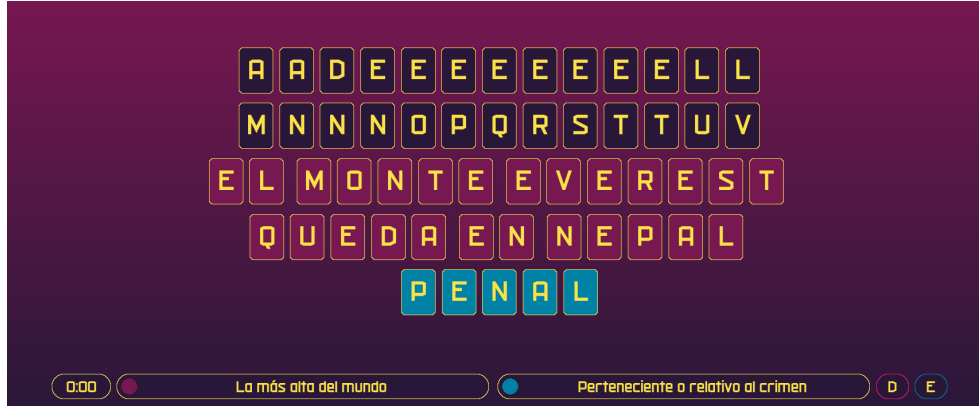


Figura 5.9: Vista de la pantalla de juego Criptocódigo

La creación del panel que vimos en la figura 5.7 podemos verlo en el panel de juego de la figura 5.9, donde vemos reflejada la información de la base de datos, acá vemos más datos que se obtienen cuando se presiona el botón *Jugar* de la figura 5.8. Vemos un temporizador en 0:00, ambas pistas vistas anteriormente, la letras menos y más repetida en el alfagrama, la frase oculta en formato alfagrama (cuadros morados), las respuestas correctas del alfagrama (cuadros fucsias) y del anagrama oculto en el alfagrama (cuadros celestes). La vista de esta forma aparece cuando se acaba el tiempo, en este momento se deben mostrar las respuestas correctas. Por último, notamos que la pantalla tiene una relación de aspecto distinta a las anteriores, esto sucede porque se realiza un *responsive* a todo el programa. Al utilizar la tecla *Shift Derecha* podemos intercambiar la vista de jugador a moderador, este último desde que comienza el juego puede ver las respuestas correctas, esto por las limitaciones que se explican en la sección 5.4 de este informe.

Para cualquier duda que se deba resolver en el juego, la figura 5.8 muestra un menú con las reglas de los cinco juegos. En particular, podemos ver las reglas del panel *Contrarreloj múltiple* en la figura 5.10. Estas reglas fueron obtenidas de las bases del juego que se encuentran en el Anexo A de este informe.

Podemos ver en la figura 5.11 los datos del usuario, el tipo de usuario, y la opción de cambiar su clave. En particular, notamos que el tipo de usuario *Administrador* que hemos tenido en las figuras anteriores cambió a *Editor* al haber hecho uso de la palabra clave. Además, podemos ver que el menú principal también sufrió cambios, ya que un editor no puede crear partidas.

Podemos encontrar más pantallas en el Anexo B de este informe.



Figura 5.10: Vista de la pantalla Reglas

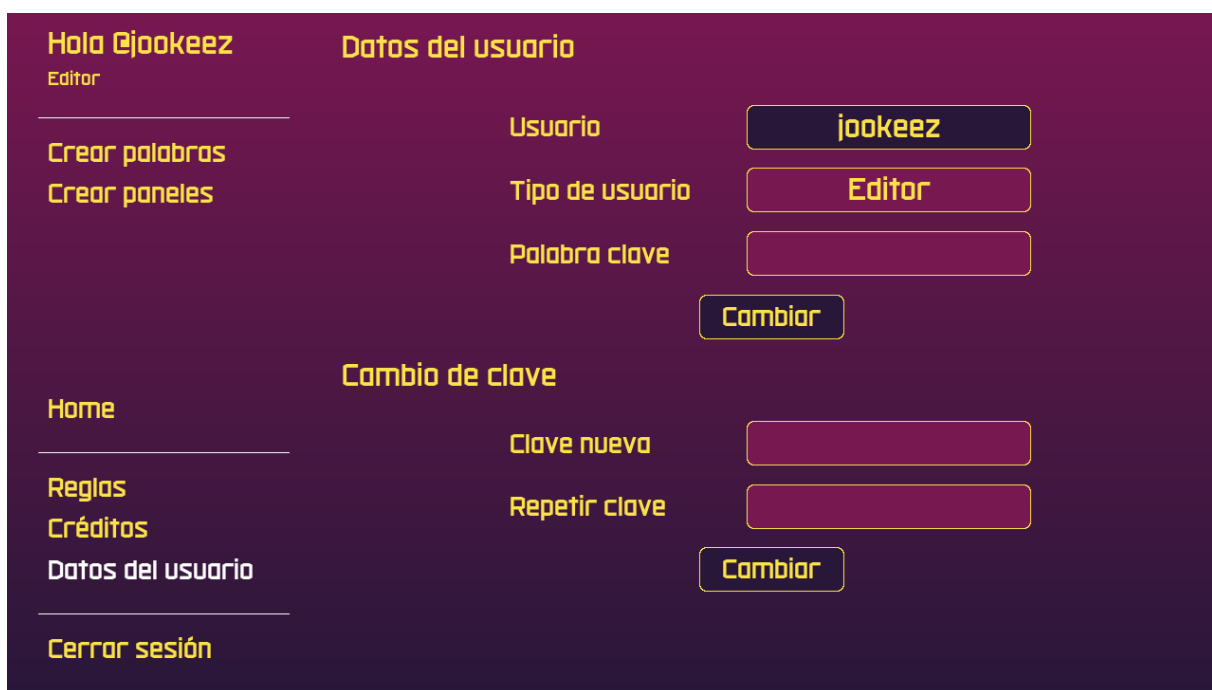


Figura 5.11: Vista de la pantalla Datos del usuario



## 5.4. Limitaciones del software

Mientras se desarrollaba el software hubo ciertos inconvenientes que impidieron una correcta ejecución de lo planificado. Uno de ellos fue la pantalla para el moderador y otra independiente para los participantes, haciendo que el moderador tuviese una pantalla con las respuestas directamente, y el participante con la pantalla de juego (como se muestra actualmente). Esto se debe a que la librería *pygame* tiene la limitante de solo ejecutar una pantalla a la vez. Para resolver esto, los paneles de moderador y jugador se intercambian con la tecla *Shift Derecha*, a pesar de que no es una solución cómoda para el usuario, el hecho de que el programa se pueda ejecutar en dos computadores soluciona rápidamente esta limitante.

Hay más limitaciones que se explican en la sección 6.1 de este informe.

## 5.5. Evaluación de algoritmos

### 5.5.1. Buscador de anagramas

En la tabla 5.1, se presenta el tiempo de ejecución promedio (en milisegundos) al realizar la búsqueda utilizando las primeras dos mil palabras del leuario como ejemplo. En los casos en que la longitud del listado es menor, se emplea la propia longitud del listado. En la figura 5.12, se observa que, a medida que la longitud del leuario se incrementa, el tiempo de búsqueda también experimenta un aumento, ya que implica una exploración más extensa.

A continuación, se presenta el pseudocódigo correspondiente a la búsqueda de anagramas en el leuario.

```
1 Algorithm buscador_anagrama(anagrama):
2   Input   anagrama a comparar
3   Output  listado de anagramas encontrados en el leuario
4
5   diccionario <- leer_diccionario(len(anagrama))
6   frecuencia_anagrama <- Counter(anagrama)
7   listado_encontradas <- []
8   for palabra_comparar in diccionario:
9       if frecuencia_anagrama es igual a Counter(palabra_comparar):
10          if palabra_comparar es distinta a anagrama:
11              agregar palabra_comparar a listado_encontradas
12   return listado_encontradas
```

Notar que, en el bucle *for*, se revisa cada palabra del diccionario, su tiempo de ejecución es  $O(m)$ , donde  $m$  es la longitud del leuario. Al utilizar el objeto *Counter* (función de python que cuenta las frecuencias de cada letra de una palabra), su tiempo de ejecución es  $O(n)$ , donde  $n$  representa la cantidad de letras de una palabra. La comparación de los objetos *Counter*, correspondientes a la palabra original y la obtenida en el leuario, tiene una complejidad de tiempo  $O(n)$ .

El algoritmo tiene una complejidad de tiempo  $O(n \cdot m)$ , donde  $n \leq 25$  y  $m \leq 15000$ . No se incluye el tiempo de ordenamiento del listado de palabras encontradas, ya que provienen directamente del leuario, el cual ya se encuentra ordenado. En la figura 5.13, se observa claramente la relación lineal del tiempo, a medida que la longitud del leuario aumenta.

Letras	Tiempo promedio (ms)	Longitud leuario
4	5.205	2075
5	13.238	5231
6	20.403	9041
7	29.264	12528
8	33.940	13968
9	33.084	13253
10	28.118	10749
11	20.230	7446
12	11.554	4753
13	8.738	2960
14	4.350	1725
15	3.269	997
16	1.811	502
17	0.942	239
18	0.563	115
19	0.249	40
20	0.142	21
21	0.099	5
22	0.078	2

Tabla 5.1: Resultados de tiempo de ejecución promedio en milisegundos

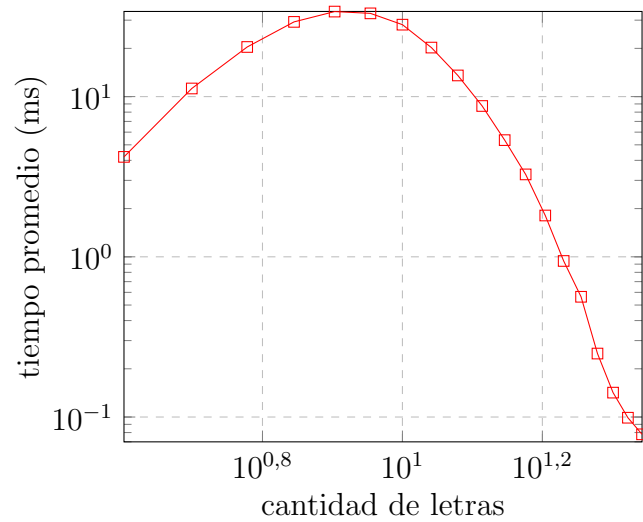


Figura 5.12: Resultados de tiempo de ejecución promedio en milisegundos. Gráfico log-log

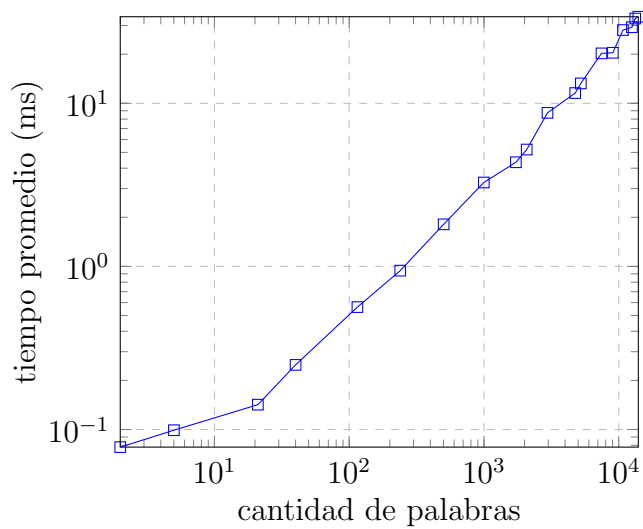


Figura 5.13: Resultados de tiempo de ejecución promedio en milisegundos. Gráfico log-log

# Capítulo 6

## Testing

Cuando nos enfrentamos a desafíos como la creación de un software, es imprescindible que este sea sometido a constantes pruebas dentro del desarrollo, esto no significa que cuando se está desarrollando no se encuentren errores a futuro, todo lo contrario, el constante cambio en el código involucra que puedan aparecer ciertos errores que no se consideraron anteriormente o que se vuelvan a realizar algunos ajustes. En este capítulo se debe enfatizar en dos fases: *alfa* y *beta*, cada una tiene un proceso propio de pruebas con distintos actores quienes realizan ciertas operaciones en el software desarrollado para buscar errores que puedan aparecer, estas fases se realizan antes de que el software pase a producción para el público en general. Quienes llevan a cabo la *fase alfa* son desarrolladores, evaluadores y analistas de calidad quienes trabajan en la empresa que desarrolla el propio software, mientras que la *fase beta* involucra a usuarios reales que evalúan el software proporcionando comentarios sobre las funcionalidades y su usabilidad. Finalmente, hablaremos de la *fase de producción* y la publicación del software en su primera versión.

### 6.1. Fase Alfa: Quality Assurance

En esta fase se aplica un conjunto de actividades de evaluación para garantizar que el producto final sea de calidad, a esto se le llama aseguramiento de calidad o *Quality Assurance* (QA por sus siglas en inglés). Esto busca que el software principalmente satisfaga las necesidades del cliente. Para esto se debe diseñar un *plan de QA* [10] con los siguientes tipos de pruebas:

- *Testeo unitario*, donde se prueba que cada módulo funcione bien por separado.
- *Test de estrés*, se prueba la resistencia de la aplicación enviándole una cantidad de peticiones excesiva, buscando que colapse.

- *Test de integración*, donde los módulos probados independientemente durante el testeo unitario se acoplan y se prueban en conjunto.
- *Test funcional*, se prueba que el software ofrezca las funciones solicitadas.
- *Test de aceptación*, donde el usuario verifica que el producto satisfaga sus expectativas.

El software de *Criptograma* requiere de un plan de QA algo más simplificado tomando en cuenta que el desarrollo de todo el producto y esta fase lo realiza una sola persona. Tomando en cuenta todo esto, a continuación se detalla cada uno de los tipos de pruebas:

### 6.1.1. Testeo unitario

Para el caso de uso de la tabla 4.2, en el panel *Crear palabras*, las palabras o frases son buscadas según lo esperado como podemos ver en la figura 5.5, para las excepciones nombradas se muestra una etiqueta de texto en la parte inferior derecha al usuario con el mensaje “El largo mínimo para buscar anagramas es de 4 letras.”.

Para el caso de uso de la tabla 4.3, en el panel *Crear paneles*, las búsquedas son realizadas sin inconvenientes, incluso si agregamos ceros a la izquierda del número. Al guardar un nuevo panel, no se sabe el id en donde se guarda. No hay mensaje de confirmación cuando se guarda o edita. No se verifica en los paneles que la cantidad de letras y las mismas correspondan a su respuesta, pudiendo agregar como ejemplo “ALAS” y “SALAS” en el panel *Palabra anagrama* a pesar de ser una respuesta incorrecta.

Para el caso de uso de la tabla 4.5, en el panel *Crear partida*, se logra abrir cada panel de juego sin problemas. Si borramos múltiples id de panel de juego, notamos que el número que se agregue siempre será el primero disponible, y no el que nosotros requerimos. La vista de moderador no está completamente integrada al panel de juego *Contrarreloj múltiple* y *Criptocódigo*.

Para el caso de uso de la tabla 4.7, en el panel *Datos del usuario*, sucede lo mismo con los id del párrafo anterior, ahora con los campos *Palabra clave*, *Clave nueva* y *Repetir clave*.

Se cumple lo esperado para los demás casos de uso.

Respecto a los componentes de la plataforma, se revisa que estos se adaptaran a distintas relaciones de aspecto y escalado, estos fueron probados en pantallas de distintas dimensiones. Notamos que el contenido se adapta correctamente a la resolución nativa de la pantalla de prueba, como lo podemos ver en las figuras 5.9 y 6.1, con relación de aspecto 21:9 y 16:9 respectivamente. Para finalizar, la resolución mínima de la ventana es de 1280 por 720 pixeles respetando la relación de aspecto 16:9.



Figura 6.1: Vista del juego en una pantalla de 43 pulgadas.

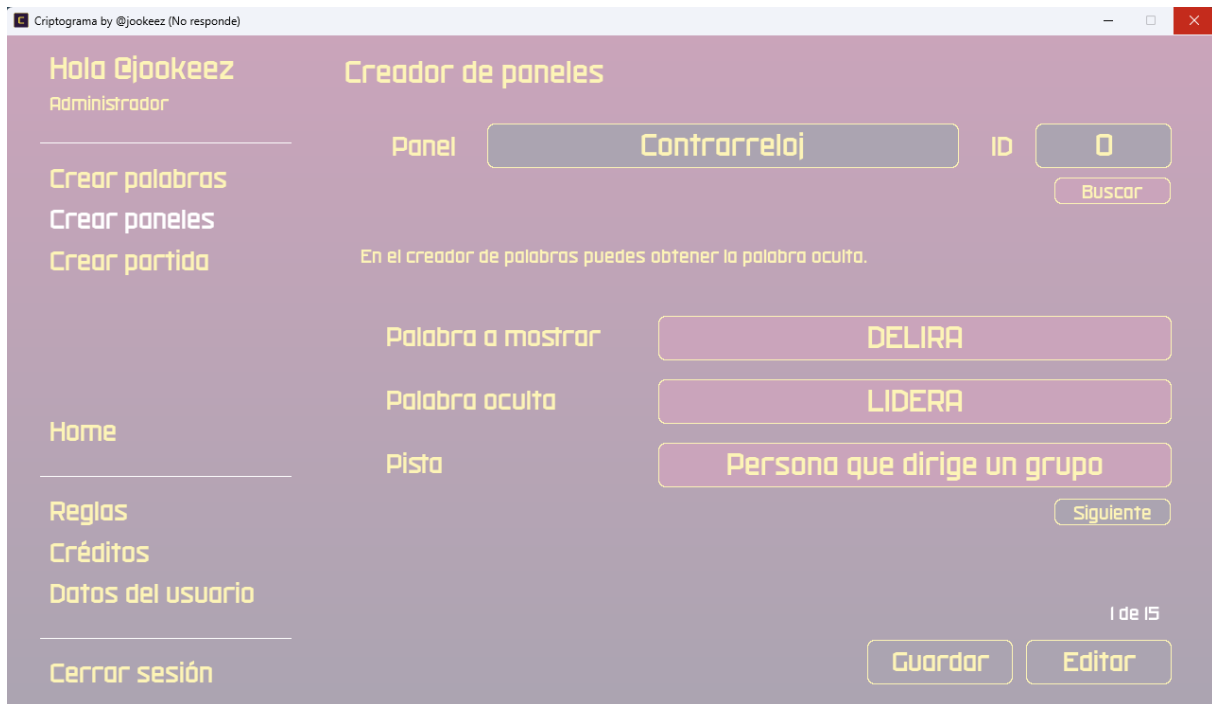


Figura 6.2: Vista de la pantalla Creador de paneles

### 6.1.2. Test de estrés

Se detecta que en *Crear paneles* al guardar la información del panel *Contrarreloj* en la base de datos, al tener que guardar la información en línea, cuando se presiona la ventana con el cursor repetidas veces, esta deja de responder por unos momentos haciendo que el sistema muestre una pantalla semi blanca como podemos ver en la figura 6.2.

Para el resto de los casos no se evidenciaron problemas.

### 6.1.3. Test de integración

Como este software no requiere de módulos de integración, no se realiza este test. Es por esta razón que la arquitectura de software utilizada solo se profundiza al nivel dos del modelo C4.

### 6.1.4. Test funcional

La única funcionalidad que no provee es la de registrar usuario como podemos ver en la figura 6.3, no existe el apartado de registro.

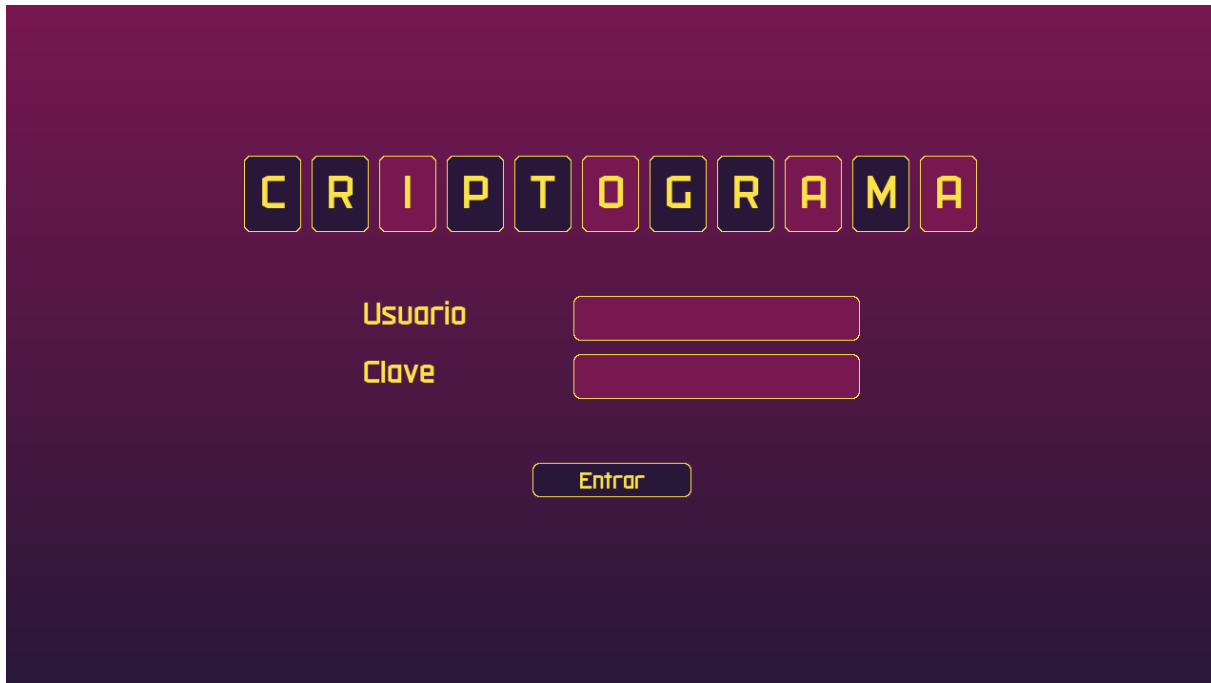


Figura 6.3: Vista de la pantalla Inicio de sesión

### 6.1.5. Test de aceptación

A pesar de los test mencionados anteriormente, es posible considerar este software como un producto mínimo viable que todavía sigue en el proceso de desarrollo, ya que es capaz de realizar funciones fundamentales tanto para el contenido como para el desarrollo del propio juego. Por último, del diseño gráfico podemos notar que la paleta de colores es visualmente atractiva, los componentes están ubicados estratégicamente para cumplir con el buen funcionamiento del *responsive* o adaptación del contenido a la pantalla.

## 6.2. Fase Beta: Validaciones de usuarios

En esta fase, es un grupo acotado de personas quienes ponen a prueba el software, para esto debemos revisar quienes son las personas más adecuadas. Considerando que el público objetivo de *Criptograma* son productoras de televisión que realizan este tipo de formato de juegos o canales de televisión, es complicado llevar a cabo una validación con las personas adecuadas quienes son los posibles usuarios. Tomando esto en cuenta, el público más adecuado para realizar estos tests son personas con conocimientos en informática y personas comunes, se decide segmentar en dos grupos simulando a los usuarios reales del juego, estos son:

- Grupo A, conformado por estudiantes, memoristas y titulados de Ingeniería Civil



Informática de la Universidad de Concepción. Estos simulan los tipos de usuario: Administrador, Moderador y Editor. Este grupo tiene el objetivo de revisar el programa con un pensamiento más abstracto.

- Grupo B, conformado por público en general. Estos simulan el rol de participante. Este grupo tiene el objetivo de jugar los 5 paneles de juego.

Al **grupo A** se le cede el control de la aplicación, en todo momento cuentan con supervisión para cualquier duda o inconveniente que tuviera el usuario. Se les solicita revisar e interactuar con todas las pantallas del menú principal. Al finalizar, se les solicita completar un test sobre la experiencia en el uso de la aplicación. Este decía lo siguiente:

Las siguientes preguntas están relacionadas con su experiencia en el uso de la aplicación Criptograma. Para cada una de estas, indique su grado de satisfacción con una escala de 1 a 7, siendo 1: Totalmente insatisfecho, 2: Muy insatisfecho, 3: Insatisfecho, 4: Indiferente, 5: Satisfecho, 6: Muy satisfecho, 7: Totalmente satisfecho.

- I. Grado de satisfacción de los componentes de la aplicación (menús, botones, información, etiquetas, cuadros de textos, elementos de interacción con el usuario)
- II. ¿Qué tan fácil fue ocupar el creador de palabras?
- III. ¿Qué tan fácil fue ocupar el creador de paneles?
- IV. ¿Qué tan fácil fue crear una partida de juego?
- V. ¿Se entiende la explicación de las reglas de juego?
- VI. ¿Qué tan fácil fue cambiar los datos del usuario?
- VII. ¿Es intuitivo el uso de la aplicación?
- VIII. Comentarios sobre el uso de la aplicación

Como resultado, participaron diez personas. El promedio de cada pregunta fue siempre sobre 5. Este grupo tuvo una buena aceptación del software en sí. Como se esperaba en un principio, este grupo fue bastante más crítico con respecto a las funcionalidades que pudiese haber tenido, como una pantalla de carga y confirmación de algún cambio realizado, una guía que indicara lo que se debe hacer y explicar algunas funcionalidades que parecían ocultas como el uso del portapapeles, las vistas de los paneles creados fueran separados del creador de paneles, entre otras funcionalidades más pequeñas. En resumen, como contaban con supervisión en todo momento, se logra llevar a cabo el uso de todas las funcionalidades principales del software que estos debían revisar.

Al **grupo B** se le solicita participar en al menos dos juegos, revisar la información en pantalla e interactuar con el pulsador. Al finalizar, se les pide completar un test sobre la experiencia

en su participación. Este decía lo siguiente:

Las siguientes preguntas están relacionadas con su experiencia en el uso de la aplicación Criptograma. Para cada una de estas, indique su grado de satisfacción con una escala de 1 a 7, siendo 1: Totalmente insatisfecho, 2: Muy insatisfecho, 3: Insatisfecho, 4: Indiferente, 5: Satisfecho, 6: Muy satisfecho, 7: Totalmente satisfecho.

- I. ¿Qué tan fácil fue entender un juego con la información mostrada en pantalla?
- II. ¿Se entiende la información adicional mostrada en pantalla? (lo que aparece en la parte inferior, como el tiempo, pistas, puntaje de cada participante, etc.)
- III. ¿Qué tan fácil fue la interacción con el pulsador?
- IV. ¿Volverías a jugar Criptograma nuevamente?
- V. Comentarios adicionales (opcional)

Como resultado, participaron diez personas. El promedio de cada pregunta fue siempre sobre 6. Este grupo tuvo muy buena aceptación con el juego en general. Los paneles que requerían de pulsador se entendieron a los pocos segundos, la información se entendía sin necesidad de explicaciones adicionales y todos estuvieron dispuestos a jugar este tipo de paneles. De los paneles con alfagramas por su complejidad a pesar de que se entiende el concepto y la forma de resolver, como requería de mayor concentración y la dificultad igual era más elevada, el público prefiere los paneles con anagramas. Los comentarios fueron repetitivos al mencionar mejoras visuales al momento de presionar el pulsador, que hubiese información en pantalla sobre quién debe responder.

### 6.3. Fase de Producción

Tomando en cuenta los resultados de ambas fases y aplicando las correcciones que son necesarias además de tener que volver a someter un nuevo proceso de pruebas con dichos cambios, se estima que durante el mes de marzo de 2024 el juego pueda entrar en producción en su primera versión.

# Capítulo 7

## Conclusiones

El proceso de desarrollo realizado ha permitido la obtención de una primera versión funcional del juego *Criptograma*. Cabe destacar que durante los ensayos con público general, este pudo desarrollarse sin mayores inconvenientes y con una muy buena aceptación.

La incorporación del buscador de anagramas al software hace que el proceso de creación de paneles sea más rápido de realizar para el editor, ya que los recursos son propios del software y no se requiere de aplicaciones externas para su búsqueda, además de añadir valor al producto. También, destacar que los tiempos de ejecución del buscador, al ser menores a un segundo inclusive en palabras grandes, hace que el trabajar con esta herramienta acorte el proceso de creación del propio contenido de juego.

Se identifican algunos elementos que conforme avanzaba el transcurso de esta memoria de título, sufrieron modificaciones como lo fue priorizar el desarrollo del buscador de anagramas. Esto significa dejar algunas funcionalidades, como el creador de partidas por identificador, guardado masivo de información en la creación de paneles, la seguridad de la aplicación, entre otras que se explicarán más adelante.

Por último, varios de los comentarios recibidos por el grupo A de validaciones, se tomaron en cuenta para una nueva actualización de este software, como separar las vistas en la creación de paneles entre guardar y editar, previsualizar los paneles creados, botones de ayuda rápida, intercambiar palabras entre los distintos paneles de juego, identificador del cursor cuando se escribe en un campo de texto, entre otras indicaciones.

## 7.1. Trabajo futuro

Llevar el software a otros idiomas requerirá de una revisión gramatical del idioma que se quiera incorporar, además de que sus letrados sean válidos en sus idiomas respectivos.

Cuando se realiza el análisis de requerimientos, habían ciertas mejoras que se decidieron no incluir para que el desarrollo del buscador de anagramas tuviese prioridad, como lo era, ordenar los paneles por categoría en el panel de juego *Contrarreloj* para que el participante tenga mayor claridad en sus respuestas, un creador de partidas que se guarde por *id* de partida en la base de datos, para que el moderador pueda tomar decisiones y cambie la partida completa antes de comenzar el juego (de ser necesario), un historial de partidas con información relevante como el tiempo jugado parcial y total, puntaje obtenido de los jugadores en cada panel, realizar estadísticas en base al comportamiento de los jugadores en los paneles como el tiempo de respuesta de cada panel resuelto correctamente, la capacidad de jugar en línea (solo como herramienta de práctica).

Actualmente el software requiere de varias correcciones que se deben mejorar para que el usuario pueda interactuar sin problemas, como la escritura tipo párrafo (no la provee *pygame*), mejoramiento del portapapeles, verificar el correcto ingreso del *id* al panel de juego en la pantalla *Crear partida*, generar un listado en PDF con las respuestas de los paneles a jugar, entre otras correcciones menores.

El repositorio de Github con el código del software lo podemos encontrar en el siguiente enlace <https://github.com/jookeez/Criptograma>

# Bibliografía

- [1] WWBM, “Rules of the Game Who Wants to Be a Millionaire? - US version”, (s.f.). [Online]. Disponible en: <https://wwbm.com/rules>
- [2] Wikiwand, “El juego de los Anillos. Modo de juego”, (s.f.). [Online]. Disponible en: [https://www.wikiwand.com/es/El\\_juego\\_de\\_los\\_anillos](https://www.wikiwand.com/es/El_juego_de_los_anillos)
- [3] Guinness World Records, “First television game show”, (s.f.). [Online]. Disponible en: <https://www.guinnessworldrecords.com/world-records/first-television-game-show>
- [4] Hmong, “Wheel of Fortune (American Game Show)”, (s.f.). [Online]. Disponible en: [https://hmong.es/wiki/Wheel\\_of\\_Fortune\\_\(American\\_game\\_show\)](https://hmong.es/wiki/Wheel_of_Fortune_(American_game_show))
- [5] Wikiwand, “5 Golden Rings”, (s.f.). [Online]. Disponible en: <https://www.wikiwand.com/nl/5%20Golden%20Rings>
- [6] Real Academia Española. Diccionario de la lengua española, 23ra. ed., Barcelona: Espasa Libros S.L.U., 2014, p. 139
- [7] Real Academia Española. Diccionario de la lengua española, 23ra. ed., Barcelona: Espasa Libros S.L.U., 2014, p. XI
- [8] Ismael Olea, “Lemarios y listas de palabras del español”, (s.f.). [Online]. Disponible en: <https://olea.org/proyectos/lemarios/>
- [9] Youtube, “What I’ve been working on - Neon Horizon Reveal”, (2021). [Online]. Disponible en: <https://www.youtube.com/watch?v=aqRiBqNM4fg>
- [10] 4rsoluciones, “¿Qué es un plan de QA?”, (2012). [Online]. Disponible en: <https://www.4rsoluciones.com/blog/que-es-un-plan-de-qa-2/>

# Anexos

# Anexo A

## Diseño de gráficas

El diseño de las gráficas del juego fue realizado con el software Adobe Illustrator. El detalle de los recursos utilizados para la creación se detalla a continuación:

- **Tamaños**

- **Ventana:** 1920px por 1080px nativo con una relación de aspecto 16:9, escalable a la pantalla del usuario. Debe respetar un *padding* horizontal de al menos 122px a cada lado de la ventana.
- **Tarjetas:** 125px por 165px para los anagramas y 65.5px por 86.5px si son combinaciones de anagramas; 90px por 119px para los alfagramas, frase oculta y anagrama oculto.
- **Contenedores inferiores:** En el eje x se adaptan al contenido de la letra, a excepción de la pista que se adapta al espacio disponible. En el eje y, 67px para todo el contenido.
- **Tipografía:** 121pt para los anagramas y 63.46pt si son combinaciones de anagramas; 87.31pt para los alfagramas, frase oculta y anagrama oculto; 48pt para la información de la parte inferior.

- **Tipografía**

- **Fuente:** Militech.
- **Estilo:** Regular.

- **Paleta de colores**

- **Background:** Gradiente lineal vertical utilizando los colores #76174F y #1A1029.

- **Cuadros:** #291739, #781851 y #0081A6.
- **Bordes:** #FBE347, #E71D68, #0081A5, #1FA637 y #C87413.
- **Letras y textos:** #FAE248.

Algunas muestras de diseño las podemos ver en las siguientes 2 figuras. El texto y contenedores en naranja son explicativos y no se incluyen en el diseño original.

**Anagrama**

L	L	E	N	A	B	A
—	—	—	—	—	—	—

**Palabra oculta**

**Tiempo**

0:29

**Pista**

Podría ser un cetáceo bastante grande

**Puntaje de cada participante**

2 6 4 1

**Alfagrama**

A	A	—	—	—	—	—	—	—	—	L	L	
M	N	N	N	O	P	Q	R	S	T	T	U	V

E	—	—	—	—	—	E	E	—	E	—	E	—	—
—	—	E	D	—	E	—	—	E	—	—	—	—	—

**Tiempo**

10:00

**Pista de la frase oculta**

La más alta del mundo

**Pista del anagrama oculto**

Este es uno de ellos

**Frase oculta**

Letra menos repetida    Letra más repetida

D    E



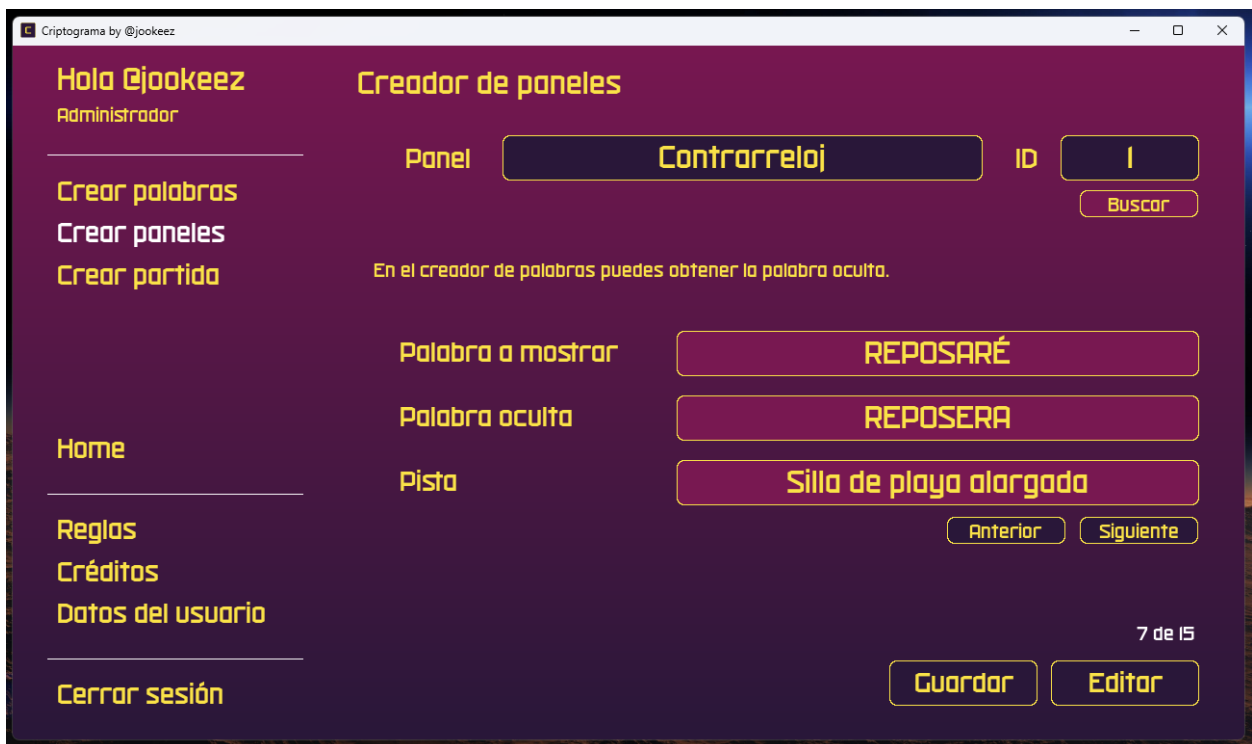
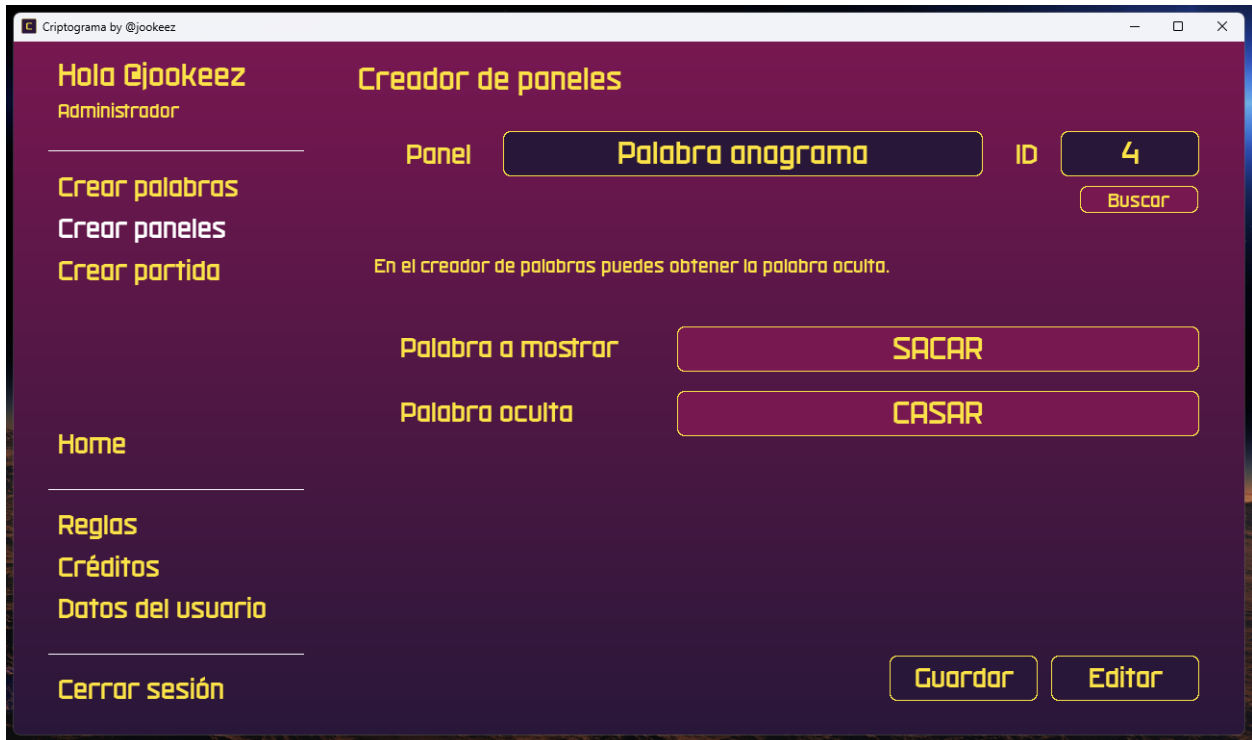
# Anexo B

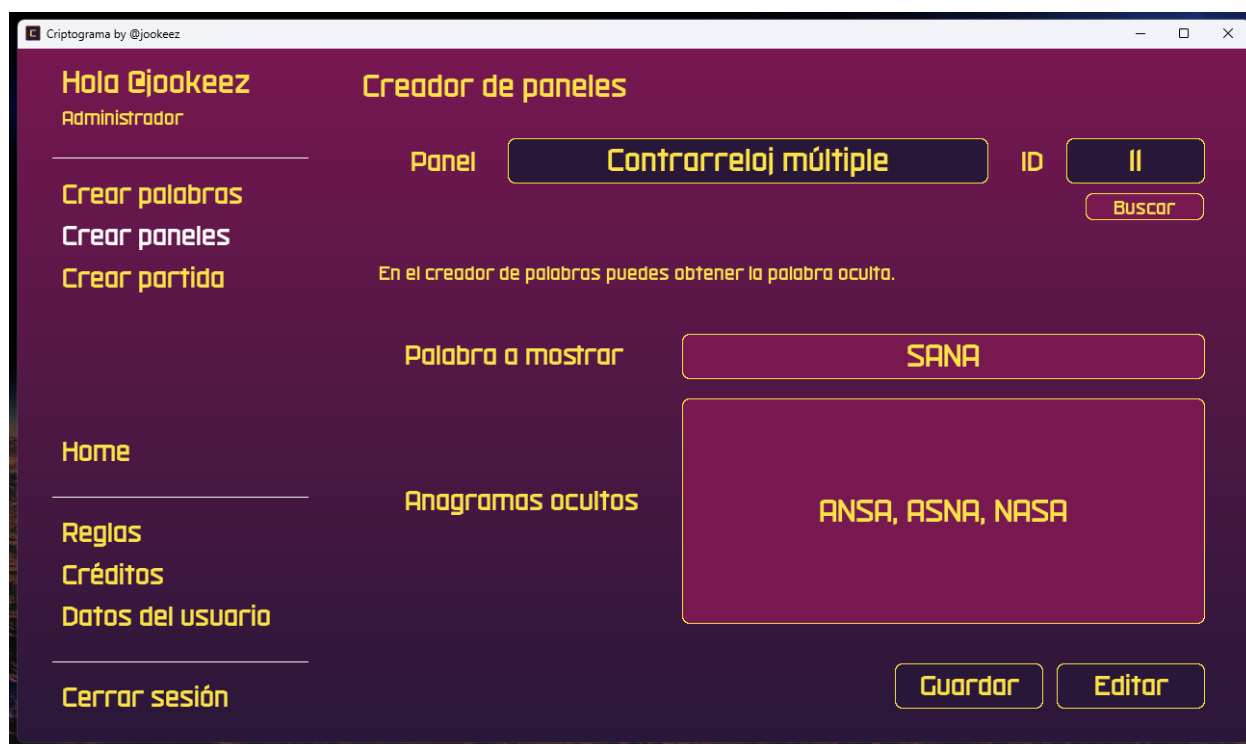
## Entorno gráfico

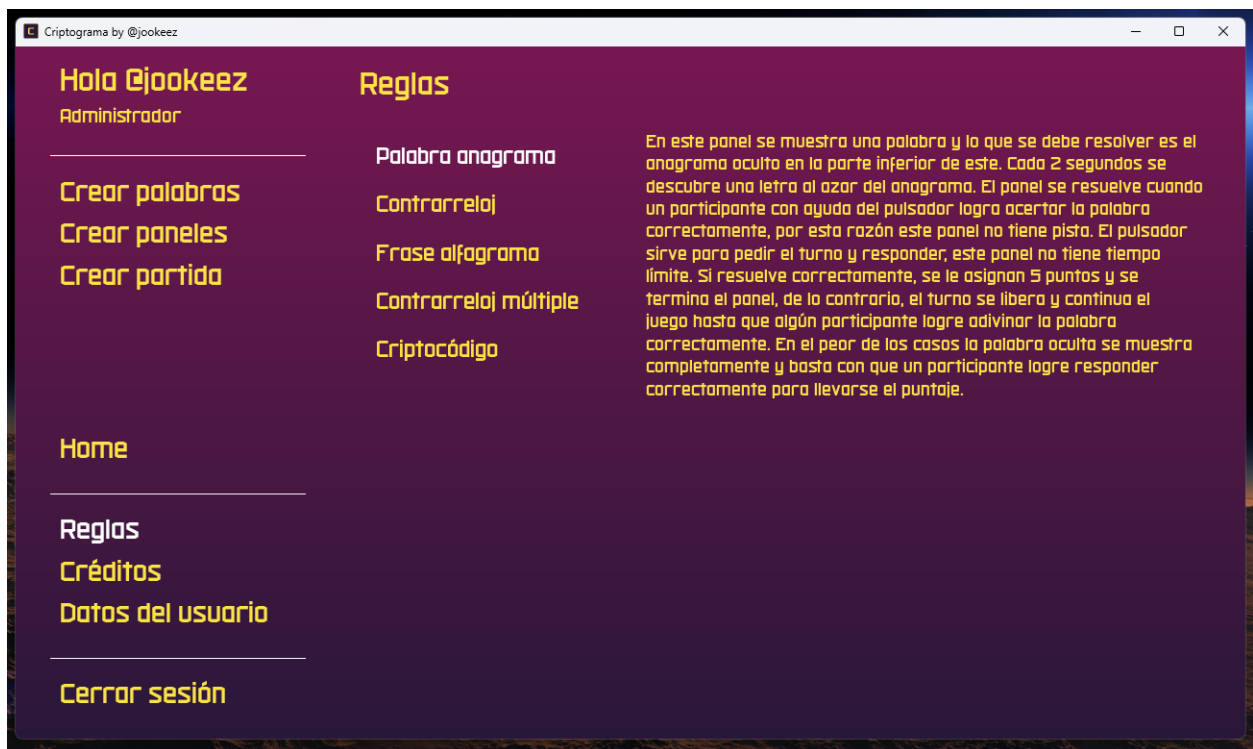
Esta es una recopilación de las pantallas diseñadas en Adobe Illustrator que fueron replicadas en el entorno gráfico del software.

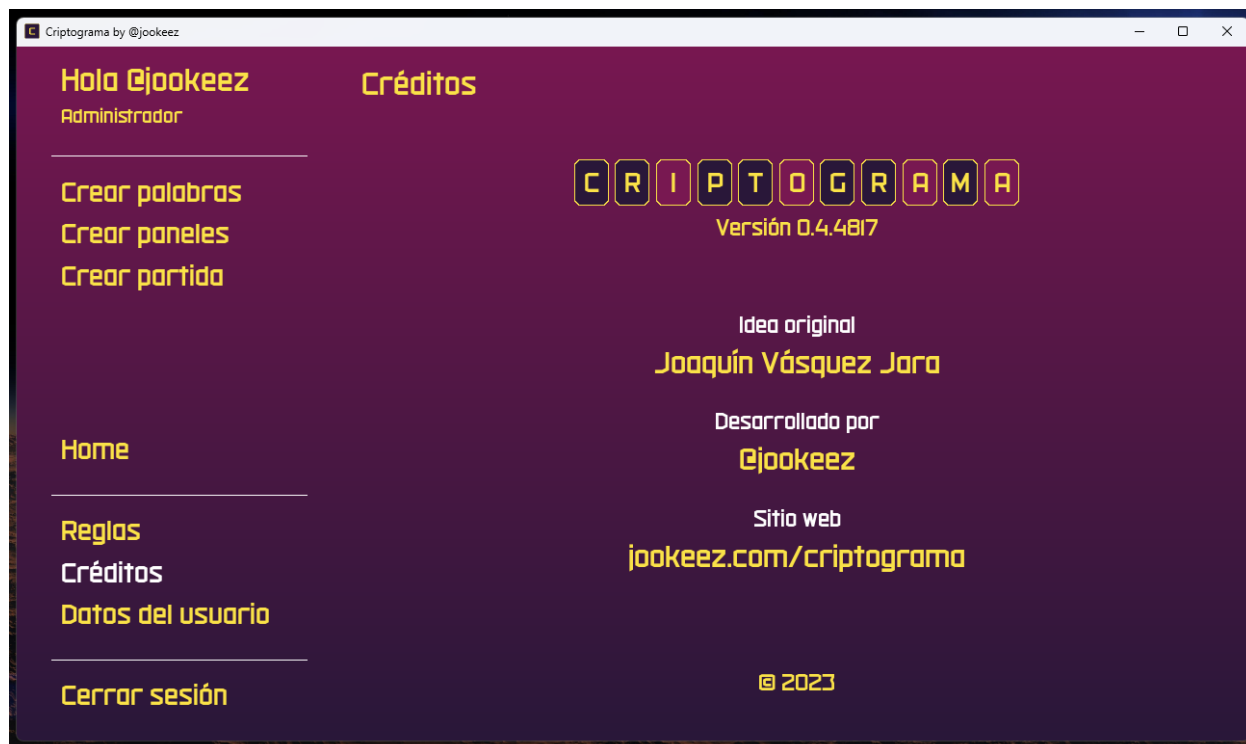


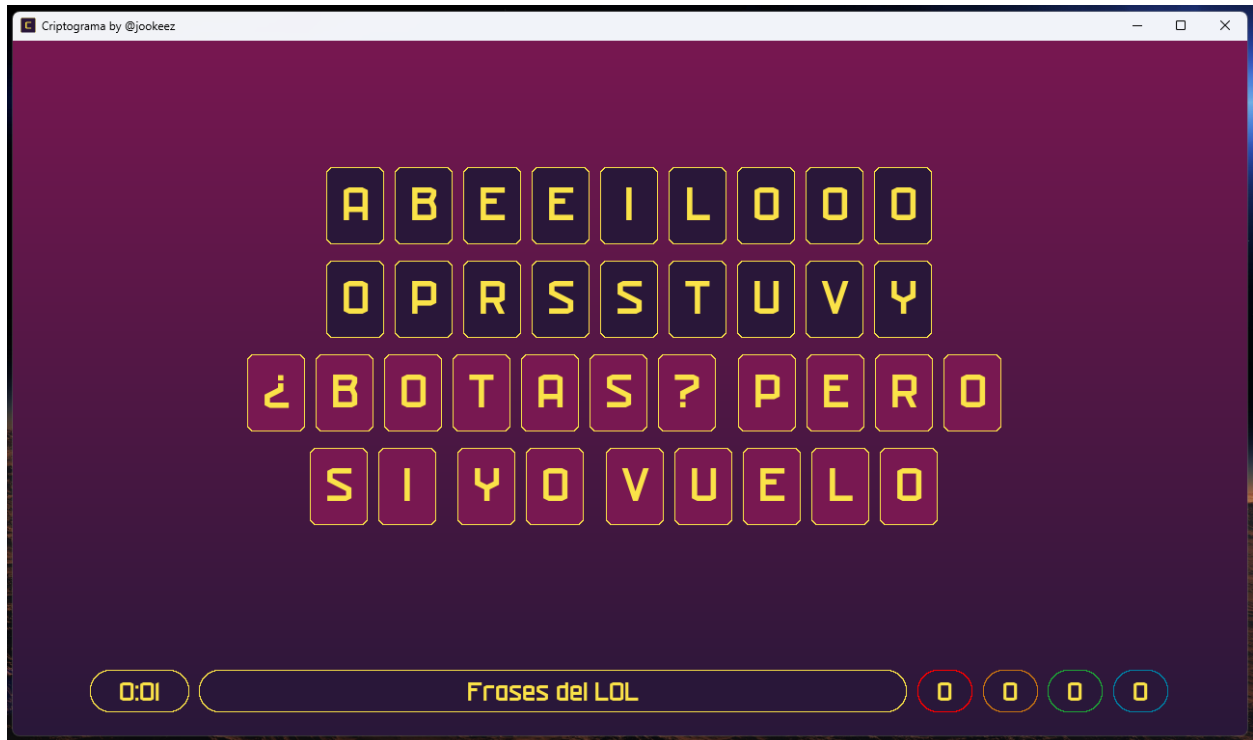
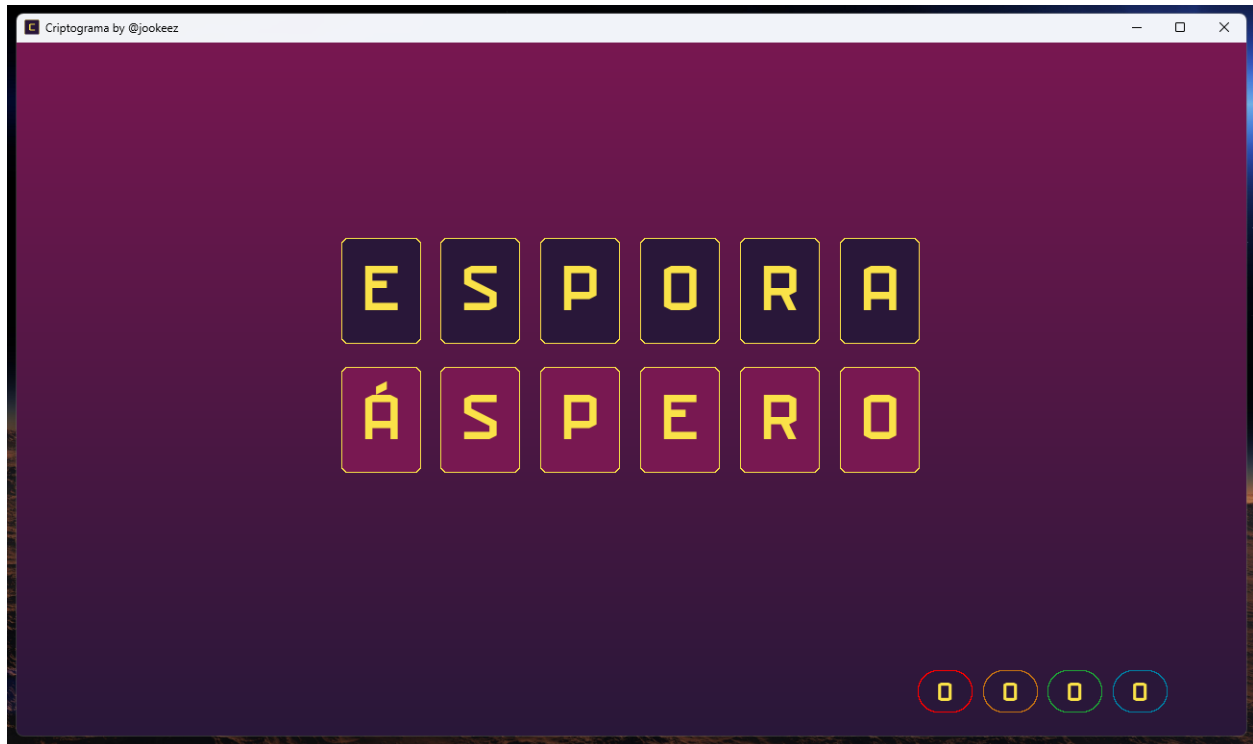












Criptograma by @jookeez

R O S A L Í A  
S A L A R I O

0:24 4 Remuneración o pago mensual 1 0 2 0

Criptograma by @jookeez

A M A R R E  
A M E R A R  
M A R E A R  
R A M E A R  
R A M E R A  
R E A M A R

0:01 0 0 0 0



Criptograma by @jookeez

A A A A A A A A B B D D D D E E E E  
E E G H I I I L L N N O P R R S T T X  
D E B Í H A B E R E X T E N D I D O  
L A G A R A N T Í A D E  
L A E S P A D A  
P E S A D A

0:07  Frase de Riven  Aburrido, que no tiene interés  G  A