



UNIVERSIDAD PERUANA
CAYETANO HEREDIA

UNIVERSIDAD PERUANA CAYETANO HEREDIA
FACULTAD DE CIENCIAS E INGENIERÍA ALBERTO CAZORLA TALLER
DEPARTAMENTO ACADÉMICO DE INGENIERÍA



TEMA:

Enunciado Trabajo Parcial: Juego “2048”

CURSO:

Base de Datos

DOCENTE:

Peter Montalvo García

ALUMNOS:

Mejia Lopez , Leonardo Camilo

Siccha Huayanay, Valery Krystal

More Ayay, Dahayra Xiomara

Huarcaya Pumacayo, Victor Nikolai

Saldaña Rodriguez, Sebastian Antonio

2024-II

Introducción

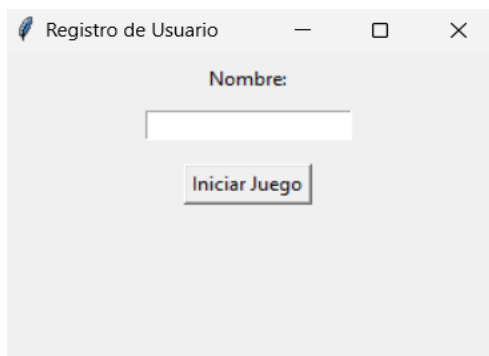
El juego 2048 ha sido escogido como base para este proyecto debido a su simplicidad y popularidad en la enseñanza de conceptos básicos de programación, ya que la funcionalidad del juego trata sobre llegar a un score alto, teniendo en cuenta los mejores movimientos que se puedan hacer para llegar al objetivo. No obstante, para aumentar la complejidad y brindar una experiencia más colaborativa, hemos implementado una dinámica de votación que permite a múltiples usuarios influir en los movimientos del juego en tiempo real. Este enfoque fomenta el trabajo en equipo y la toma de decisiones colectivas, lo que nos permite abordar retos técnicos como la integración de bases de datos, la sincronización en tiempo real.

1. Avances Realizados

Hasta el momento, nuestro grupo a logrado completar los siguientes hitos del proyecto:

- **Desarrollo del registro de jugadores:** Hasta la fecha, se ha logrado implementar una versión funcional del juego 2048 con una dinámica de votación en tiempo real. Se ha desarrollado la primera ventana (Ventana 1) que incluye los botones de control para los movimientos del juego: Arriba, Abajo, Izquierda y Derecha. Además, la segunda ventana (Ventana 2) muestra el juego sin los controles, siguiendo las instrucciones del enunciado, mientras que la tercera ventana muestra el puntaje, controles y el tiempo de juego.

Ventana 1

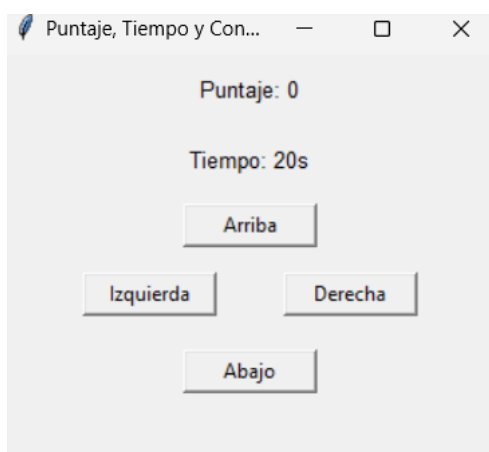


Registro de Usuario

Nombre:

Iniciar Juego

Ventana 3



Puntaje, Tiempo y Con...

Puntaje: 0

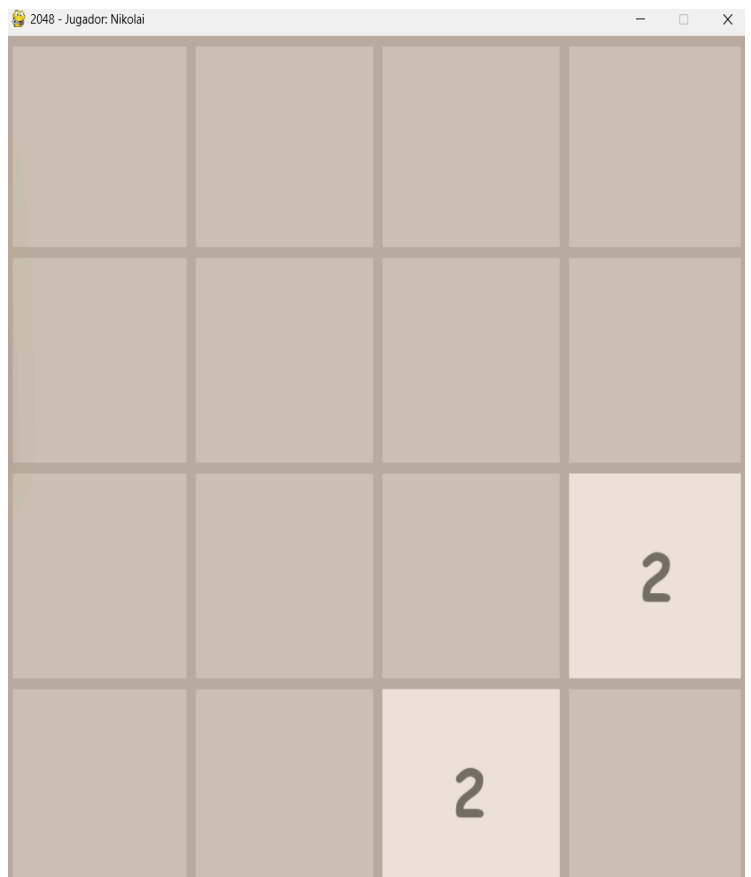
Tiempo: 20s

Arriba

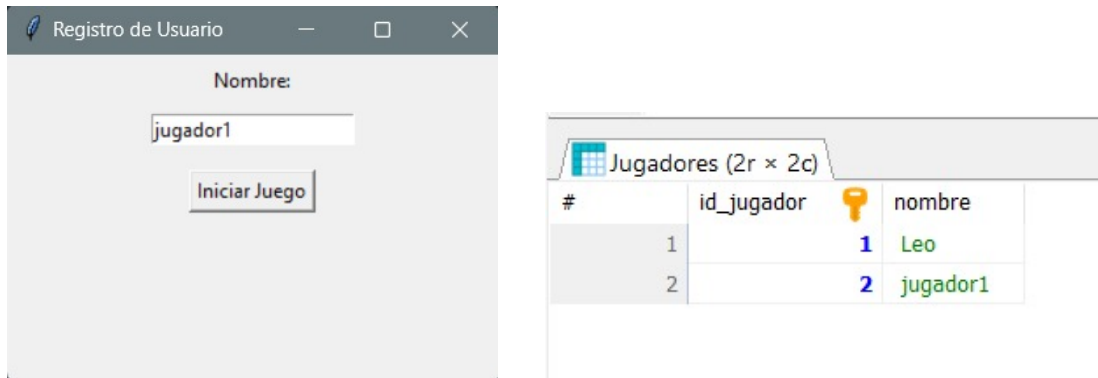
Izquierda Derecha

Abajo

Ventana 2



- **Conexión con base de datos:** Se estableció una conexión con la base de datos **DB_PARCIAL**, la cual gestiona los jugadores y los votos emitidos. La conexión a la base de datos en este proyecto es esencial para almacenar y procesar los votos emitidos por los jugadores.



The image shows two side-by-side screenshots. The left screenshot is a window titled 'Registro de Usuario' with a text input field labeled 'Nombre:' containing the text 'jugador1' and a button labeled 'Iniciar Juego'. The right screenshot shows a database table view titled 'Jugadores (2r x 2c)' with the following data:

#	id_jugador	nombre
1	1	Leo
2	2	jugador1

Almacenamiento del Jugador en la base de datos

Cada vez que un jugador selecciona una opción (Arriba, Abajo, Izquierda, Derecha), este voto es enviado a la base de datos, donde se almacena en una tabla junto con la información del jugador y el tiempo del voto. Luego, un procedimiento almacenado contabiliza los votos y determina la opción ganadora. Una vez procesados los votos, estos se marcan como leídos mediante una operación de actualización. Finalmente, el movimiento ganador se refleja en ambas ventanas del juego.





- **Implementación de la dinámica de votación:** Los jugadores pueden emitir votos para los movimientos del juego (Arriba, Abajo, Izquierda o Derecha). Estos votos son almacenados en la tabla **Votos** de la base de datos, y un procedimiento almacenado contabiliza los votos y determina el movimiento ganador.

Nombre: Votos

Comentario:

Columnas:

Agregar

Borrar

Subir

Bajar

#	Nombre	Tipo de datos	Longitud/Conjunto	Sin signo	Permitir ...	Rellen...	Predeterminado
1	id_voto	INT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME...
2	id_jugador	INT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
3	direccion	VARCHAR	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
4	timestamp	DATETIME		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	CURRENT_TIMEST...
5	leido	TINYINT	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0

Vista del apartado Votos de la Base de Datos

	id_voto	id_jugador	direccion	timestamp	leido
15	15	1	Arriba	2024-10-01 21:56:12	1
16	16	1	Arriba	2024-10-01 21:56:13	1
17	17	1	Arriba	2024-10-01 21:56:16	1
18	18	1	Abajo	2024-10-01 21:56:20	1
19	19	1	Izquierda	2024-10-01 21:56:23	1

Vista del almacenamiento de Votaciones

Una vez que el tiempo de votación ha finalizado, un procedimiento almacenado en la base de datos se encarga de contabilizar los votos, determinar la dirección ganadora y resolver cualquier empate de manera aleatoria. Posteriormente, el movimiento ganador se aplica en el juego, y las ventanas del mismo se actualizan automáticamente para reflejar el nuevo estado del tablero.

```
Run main x
Hello from the pygame community. https://www.pygame.org/contribute.html
Jugador jugador1 registrado en la base de datos.
Voto registrado: Arriba
Voto registrado: Arriba
Voto registrado: Arriba
Voto registrado: Abajo
Voto registrado: Izquierda
Movimiento ganador: Arriba
```

Resultado en la terminal de Pycharm



A 4x4 grid representing the initial game state. The grid is mostly empty, with the number '2' in the top-right cell (row 1, column 3) and the bottom-left cell (row 4, column 1). The window title is '2048 - jugador:jugador1'.

		2	
2			

Vista del juego inicial



A 4x4 grid representing the game state after voting. The number '2' is in the top-left cell (row 1, column 1) and the top-right cell (row 1, column 3). The number '4' is in the second row, second column (row 2, column 2). The window title is '2048 - jugador:jugador1'.

2		2	
	4		

Vista del juego después de la votación

- **Sincronización de resultados:** El procedimiento almacenado **ContabilizarVotos()** permite contar los votos, resolver empates de manera aleatoria, y marcar los votos como procesados. Esto asegura que el juego refleje siempre los últimos resultados de la votación.

La **sincronización de resultados** es fundamental para garantizar que el juego refleje de manera precisa los resultados de la votación en tiempo real. El procedimiento almacenado **ContabilizarVotos()** es clave en este proceso, ya que se encarga de:

- ☐ **Contar los votos** emitidos por los jugadores para las direcciones posibles (Arriba, Abajo, Izquierda, Derecha).
- ☐ **Resolver empates** de manera aleatoria cuando varias direcciones reciben la misma cantidad de votos, para asegurar que siempre haya un movimiento ganador.
- ☐ **Marcar los votos como procesados**, cambiando el estado de los votos a "leído" una vez que han sido contabilizados, lo que evita que los mismos votos se cuenten nuevamente.



```
1 CREATE DEFINER='root'@'%' PROCEDURE `ContabilizarVotos`()
2 LANGUAGE SQL
3 NOT DETERMINISTIC
4 CONTAINS SQL
5 SQL SECURITY DEFINER
6 COMMENT ''
7 BEGIN
8     DECLARE movimiento_ganador VARCHAR(10);
9     DECLARE max_votos INT;
10
11 -- Limpiar la tabla de resultados de votaciones
12 TRUNCATE TABLE ResultadosVotacion;
13
14 -- Insertar en la tabla de resultados el conteo de votos no leídos
15 INSERT INTO ResultadosVotacion (direccion, cantidad_votos)
16 SELECT direccion, COUNT(*)
17 FROM Votos
18 WHERE leido = 0
19 GROUP BY direccion;
20
21 -- Obtener la dirección con el mayor número de votos
22 SET max_votos = (SELECT MAX(cantidad_votos) FROM ResultadosVotacion);
23
24 -- Si hay un empate, elegir uno de manera aleatoria
25 SET movimiento_ganador = (SELECT direccion
26                             FROM ResultadosVotacion
27                             WHERE cantidad_votos = max_votos
28                             ORDER BY RAND());
```

2. Dificultades Encontradas

Durante el desarrollo del proyecto, hemos enfrentado varios desafíos técnicos:

- **Latencia en la actualización de las ventanas:**

Se encontraron problemas al intentar que las ventanas 1 y 2 se actualicen en tiempo real de manera sincronizada después de procesar los votos. A pesar de implementar señales para disparar las actualizaciones, hubo retrasos significativos entre el momento en que se procesaba el voto y la actualización del estado del juego en ambas ventanas, lo que generaba una experiencia inconsistente para los jugadores.

- **Concurrencia en la base de datos:**

La gestión de la concurrencia fue otro desafío importante, ya que múltiples jugadores emiten votos al mismo tiempo, lo que generó conflictos en la base de datos. Los problemas principales incluían la inserción simultánea de datos y la posibilidad de que se procesaran votos duplicados o se generaran bloqueos en las transacciones. Este problema comprometió la integridad de los datos y la correcta contabilización de los votos.

- **Integración de Pygame y Tkinter:**

Debido a que Pygame y Tkinter manejan bucles de eventos de manera diferente, integrar ambos entornos en una misma interfaz gráfica presentó dificultades. La combinación de las bibliotecas provocó problemas de rendimiento y dificultades para mantener una experiencia de usuario fluida. Cada intento de utilizar los dos marcos al mismo tiempo resultaba en comportamientos inesperados, como la congelación de la interfaz o la falta de respuesta del juego.



- **Almacenamiento del puntaje:**

Hubo complicaciones para almacenar el puntaje obtenido en cada intento del juego, así como para llevar un registro confiable de los intentos en la base de datos. Aunque se diseñó una tabla específica para almacenar estos datos, su correcta integración con el flujo del juego resultó ser más compleja de lo esperado, causando inconsistencias en los puntajes registrados y en el seguimiento de los intentos realizados por los jugadores.

3. Soluciones Intentadas

1. **Sincronización del Movimiento en Todos los Usuarios:**

Una de las primeras soluciones intentadas fue lograr que el movimiento generado por la votación se ejecutará de manera simultánea en todas las computadoras conectadas al sistema. Para ello, se implementó un mecanismo que actualiza el estado del juego en tiempo real en cada usuario conectado, utilizando una señal que dispara la actualización del tablero tan pronto como se decide el movimiento ganador. Esta señal fue diseñada para sincronizar todos los clientes y evitar que hubiera discrepancias entre los estados de las diferentes ventanas.

2. **Registro del ID del Emisor del Voto:**

Se intentó implementar un sistema que registrara el ID de cada jugador al momento de emitir un voto. Esto fue crucial para garantizar que se identifique quien emitió cada voto. Se diseñaron procedimientos almacenados en la base de datos que, al insertar un nuevo voto, asociaban el ID del jugador al voto emitido. Esta solución mejoró la trazabilidad de los votos

3. **Generación Simultánea del Movimiento en Cada Usuario:**

Se buscó una solución para garantizar que, al identificar el ID de cada jugador, el movimiento del juego se generará simultáneamente en todos los usuarios conectados. Esto se intentó mediante la identificación del ID del jugador en el proceso de ejecución de los movimientos, permitiendo que la base de datos gestionará los votos de cada usuario en tiempo real y enviará la acción correspondiente a todos los clientes conectados. Esta solución ayudó a que todos los usuarios vieran el mismo movimiento reflejado al mismo tiempo, mejorando la consistencia y la experiencia de juego.

4. **Optimización de la Base de Datos y Procedimientos Almacenados:**

Se trabajó en la optimización de los procedimientos almacenados que procesan los votos. Estos procedimientos se mejoraron para manejar la concurrencia y evitar colisiones de datos cuando múltiples usuarios emitían votos al mismo tiempo. Además, se implementaron actualizaciones eficientes para asegurar que, una vez procesado un voto, este quedara marcado como leído en la base de datos, minimizando el riesgo de que los mismos votos se procesarán más de una vez.



5. Generación de una Tabla de Resultados Obtenidos en Cada Intento:

Se intentó implementar una solución que registrara los resultados obtenidos en cada intento del juego. Para ello, se creó una tabla específica en la base de datos destinada a almacenar los resultados de cada partida, incluyendo datos como el ID del jugador, el puntaje alcanzado, el tiempo transcurrido, y el número de movimientos realizados. Esta tabla permite llevar un historial detallado de las partidas jugadas, facilitando el análisis de los resultados obtenidos por cada jugador y mejorando la retroalimentación del sistema. Esto no solo ayuda en el seguimiento del rendimiento individual, sino que también proporciona una base de datos que puede ser utilizada para futuros análisis estadísticos o mejoras del juego.

Referencias Bibliográficas

Marcus, M. (n.d.). *Cómo Jugar 2048 - Una Guía Completa*. Coolmathgames.com.
Retrieved October 1, 2024, from
<https://www.coolmathgames.com/es/blog/c%C3%B3mo-jugar-2048>