Universidad del Valle de Guatemala
Departamento de Matemática
Licenciatura en Matemática Aplicada

Estudiante: Rudik Roberto Rompich
E-mail: rom19857@uvg.edu.gt
Carné: 19857

CC2003 - Algoritmos y Estructuras de Datos - Catedrático: Melvin García
18 de mayo de 2021

Proyecto 2 - Sistema de recomendaciones

1. Investigación

Los algoritmos descritos a continuación son tomados del libro de Needham and Hodler (2019).

1.1. Strongly Connected Components

El SCC hace referencia a un algoritmo que encuentra conjuntos de nodos conectados en grafos directos; donde cada nodo es alcanzable en ambas direcciones desde cualquier otro nodo del mismo conjunto.

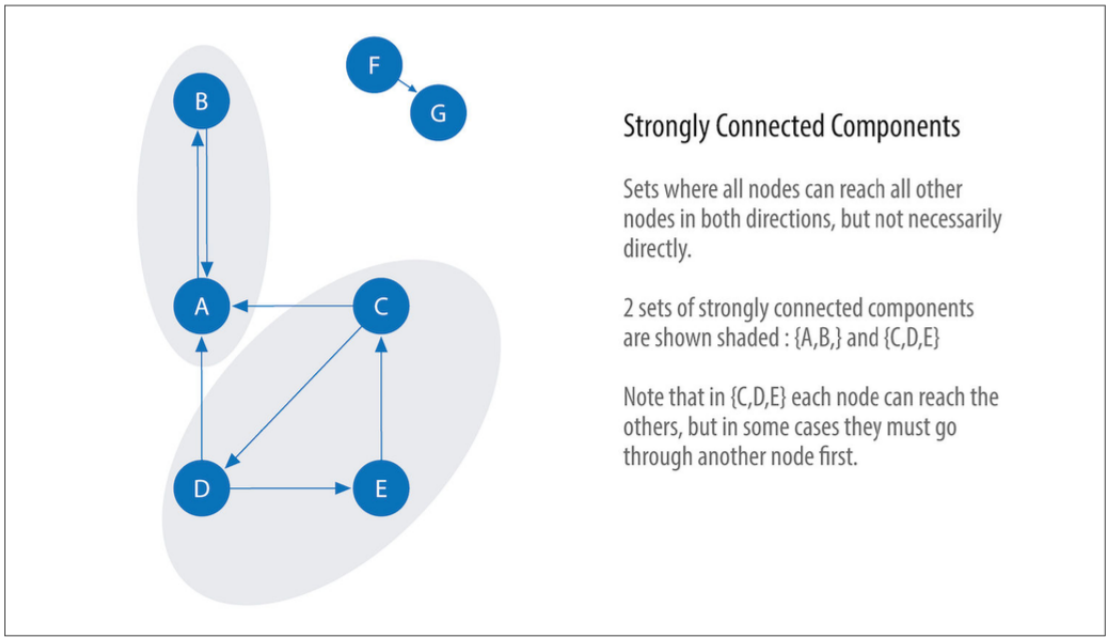


Figure 6-5. Strongly connected components

Características:

- No es necesario que los nodos sean vecinos, pero deben haber aristas entre todos los nodos del conjunto.
- Descomponer un grafo directo en un algoritmo SCC es una aplicación de otro algoritmo: *Depth First Search algorithm*.
- Un componente fuertemente conectado tiene una utilidad directa o inclinación para encontrar comportamientos similares.

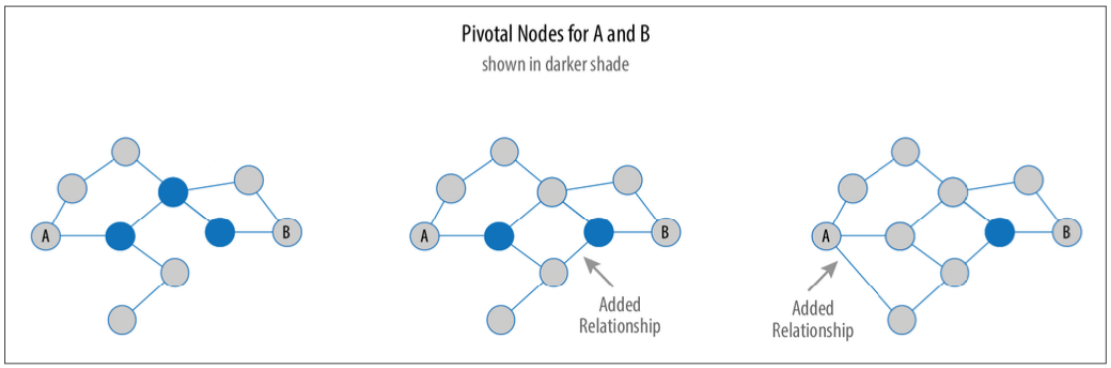
1.2. Betweenness Centrality

Este es un algoritmo que tiene como propósito detectar la cantidad de influencia que un nodo tiene sobre un flujo de información o recursos de un grafo. Una analogía sería: un algoritmo que intenta encontrar nodos que sirvan como puentes de una parte del grafo hacia otra.
Características:

- Este algoritmo primero calcula la trayectoria más corta (*weighted*) entre cada par de nodos en un grafo conectado. Cada nodo recibe un puntaje, basado en el número de trayectorias que pasan a través del nodo. Entre más trayectorias cortas pase un nodo, más alto es su puntaje.
- Puentes y puntos de control

Puente: Puede ser un nodo o una relación. Se pueden identificar, ya que si si se remueve un punte, el grafo se convierte en un grafo disconexo.

Pivote: Un nodo es considerado pivote para otros dos nodos, si ese nodo está en la trayectoria más corta de esos dos nodos. Es decir:



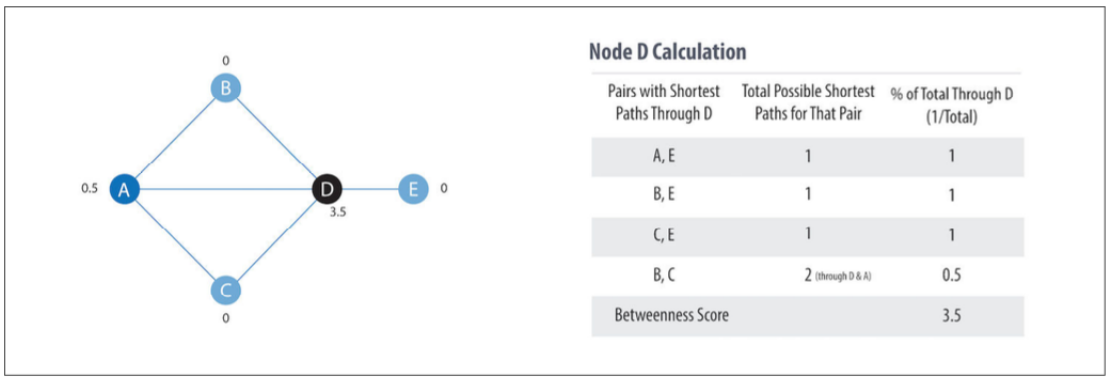
1.2.1. ¿Cómo se calcula el betweenness centrality?

$$B(u) = \sum_{s \neq u \neq t} \frac{p(u)}{p}$$

En donde:

1. u es un nodo.

2. p es el número de trayectorias cortas entre los nodos s y t.
3. $p(u)$ es el número de trayectorias cortas entre los nodos s y t que pasan a través del nodo u.



Node D Calculation

Pairs with Shortest Paths Through D	Total Possible Shortest Paths for That Pair	% of Total Through D (1/Total)
A, E	1	1
B, E	1	1
C, E	1	1
B, C	2 (through D & A)	0.5
Betweenness Score		3.5

Figure 5-8. Basic concepts for calculating betweenness centrality

Un ejemplo:

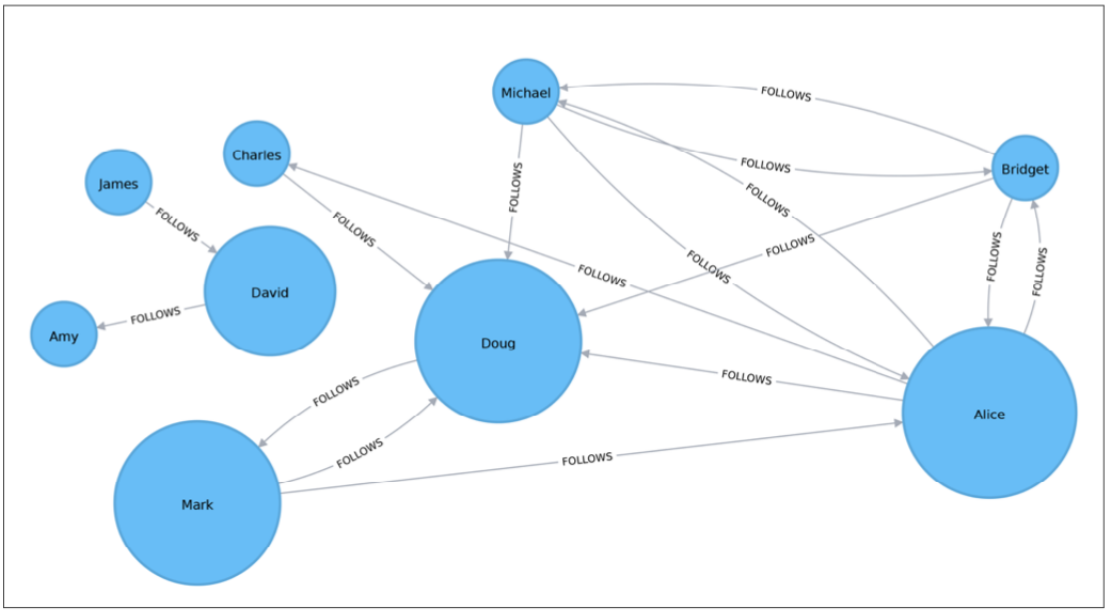


Figure 5-9. Visualization of betweenness centrality

1.3. PageRank

PageRank es el mejor algoritmo de centralidad conocido. El algoritmo mide la transitividad (o dirección) de la infuencia de los nodos. PageRank considera la influencia de los vecinos de un nodo y de sus vecinos. Por ejemplo, una analogía, una persona que tenga pocos amigos poderosos puede ser más influyente que alguien que tenga muchos amigos pero menos poderosos.

1.3.1. ¿Cómo se calcula?

Citando literalmente a Needham and Hodler (2019). «PageRank is defined in the original Google paper as follows:

$$PR(U) = (1 - d) + d\left(\frac{PR(T_1)}{C(T_1)} + \cdots + \frac{PR(T_n)}{C(T_n)}\right)$$

where:

1. We assume that a page u has citations from pages T_1 to T_n.
2. d is a damping factor which is set between 0 and 1. It is usually set to 0.85. You can think of this as the probability that a user will continue clicking. This helps minimize rank sink, explained in the next section.
3. $1 - d$ is the probability that a node is reached directly without following any relationships.
4. $C(T_n)$ is defined as the out-degree of a node T.»

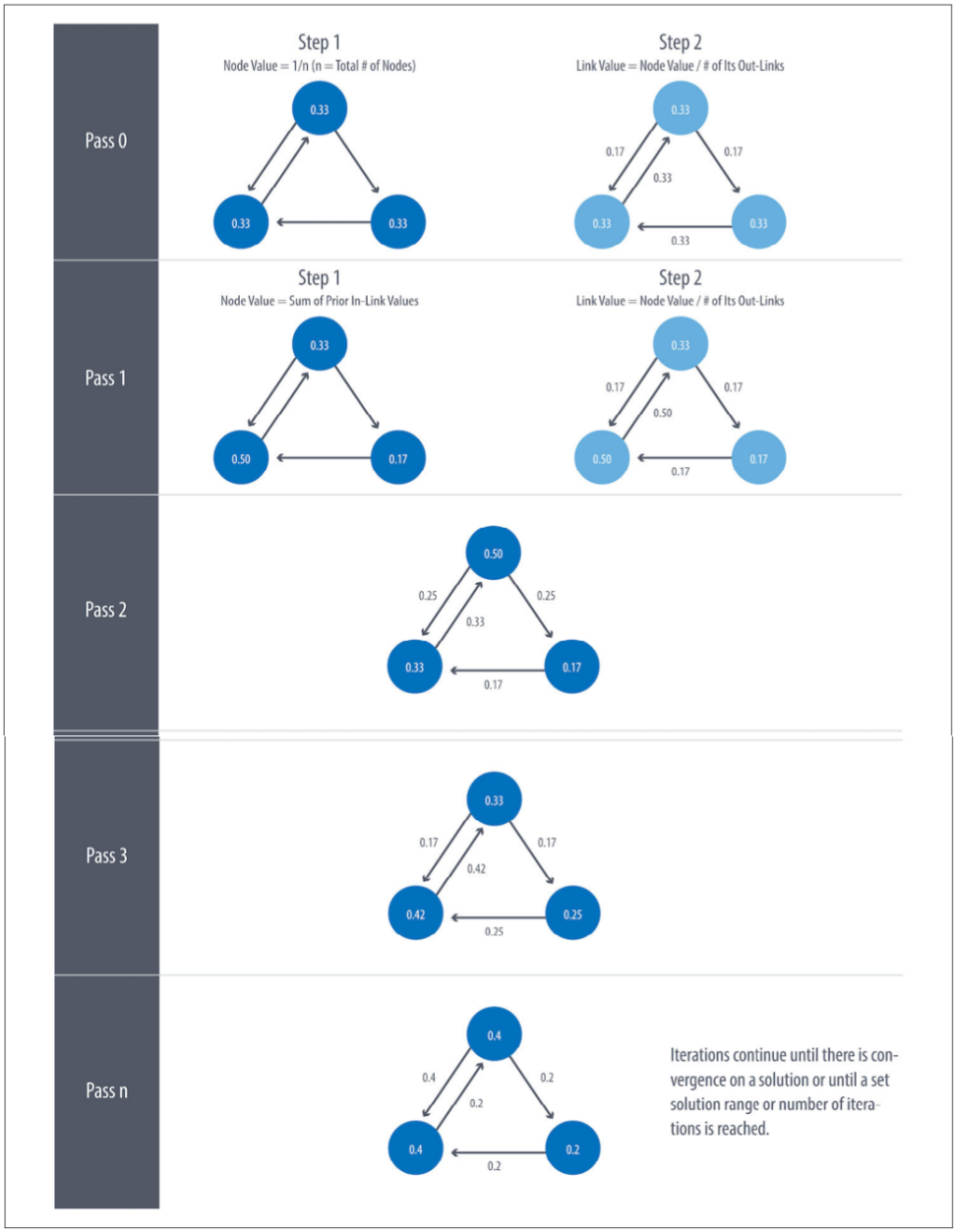


Figure 5-11. Each iteration of PageRank has two calculation steps: one to update node values and one to update link values.

2. Design Thinking

2.1. Empatía

Cada semestre en el Club de Ajedrez de la Universidad del Valle de Guatemala, entran decenas de personas esperando inmiscuirse en el mundo del ajedrez; sin embargo, se encuentran con la barrera del desconocimiento de varios principios fundamentales de este deporte y finalmente, estas personas, terminan abandonando el club o simplemente siendo miembros inactivos.

La junta directiva del Club de Ajedrez ha intentando implementar diversas actividades para evitar que los miembros se vuelven inactivos; por lo cual, se busca generar un sistema de recomendaciones que ayude a los miembros del club a tener una visión más amplia del ajedrez y que este se vuelva más envolvente en sus vidas.

2.2. Definición

Los principales problemas identificados son:

1. Desconocimiento de las fases del ajedrez.
2. Desconocimiento de las estrategias adecuadas para cada nivel de juego.
3. Herramientas adecuadas para aprender a jugar ajedrez.

El problema propuesto a resolver es un sistema de recomendaciones de cierto de tipo de posiciones.

2.3. Ideación

Existen diversas fases en el ajedrez, sin embargo, el sistema de recomendaciones se centrará en la primera fase: la apertura.
El propósito es encontrar las aperturas jugadas por los miembros de club que tengan el mismo nivel. Es decir, las 3 categorías:

1. Principiante
2. Intermedio
3. Avanzado

2.4. Prototipos

La primera fase se ha basado en elegir los parámetros más idóneos a evaluar, se han propuesto los más relevantes:

1. Plataforma favorita para jugar.
2. Plataforma con los mejores recursos para aprender.
3. Modalidad favoritas de juego.
4. Modalidad que le gusta observar en Youtube, Twitch, etc...

5. Parte favorita de una partida.
6. Su nivel de juego en Blitz.
7. Su nivel de juego en Rápidas.
8. Apertura que juega.
9. Defensa que juega.

De los cuales, únicamente se tomaron en cuenta los siguientes parámetros:

1. $PLATAFORMA$ - Plataforma favorita para jugar.
2. $APERTURA$ - Apertura que juega.
3. $DEFENSA$ - Defensa que juega.
4. $PARTE\_FAVORITA$ - Parte favorita de una partida.
5. $NIVEL\_BLITZ$ - Su nivel de juego en Blitz.
6. $NIVEL\_RAPIDAS$ - Su nivel de juego en Rápidas.

2.5. Testing

La encuesta se encuentra aquí: https://forms.gle/5swmk2VsKMPd1hha6.

La encuesta se basó en las siguientes preguntas:

1. Nivel de juego en blitz (*Considere un estimado de su rating en chess.com: (1) Principiante [0 a 1400 elo], (2) Intermedio [1400 - 1600 elo], (3) Avanzado [1600- infinito].*):
   a) Principiante
   b) Intermedio
   c) Avanzado
2. Nivel de juego en rápidas (*Considere un estimado de su rating en chess.com: (1) Principiante [0 a 1400 elo], (2) Intermedio [1400 - 1600 elo], (3) Avanzado [1600- infinito].*):
   a) Principiante
   b) Intermedio
   c) Avanzado
3. ¿Cuál es tu parte favorita del ajedrez?
   a) Apertura
   b) Intermedio
   c) Final

4. ¿Cuál es tu plataforma favorita para jugar?
   a) Lichess.org
   b) Chess.com
   c) Chess24.com
   d) Otro
5. De las aperturas anteriores, ¿cuál se adecúa más a tu estilo de juego?
   a) Italiana/Española
   b) Inglesa
   c) Sistema Londres
   d) Fianchetto
6. De las defensas anteriores, ¿cuál se adecúa más a tu estilo de juego?
   a) Eslava
   b) Francesa
   c) Caro-Kann
   d) Siciliana

2.5.1. Explicación

Por el planteamiento del problema, la idea es que el usuario ingrese su nivel de juego (principiante, intermedio o avanzado). Entonces el programa le recomendará la plataforma, apertura y defensa de jugadores de su mismo nivel. Desde el punto de vista matemático, el programa únicamente buscará las relaciones y nodos conectados a los niveles de juego.

Las aperturas y defensas en las siguientes páginas.

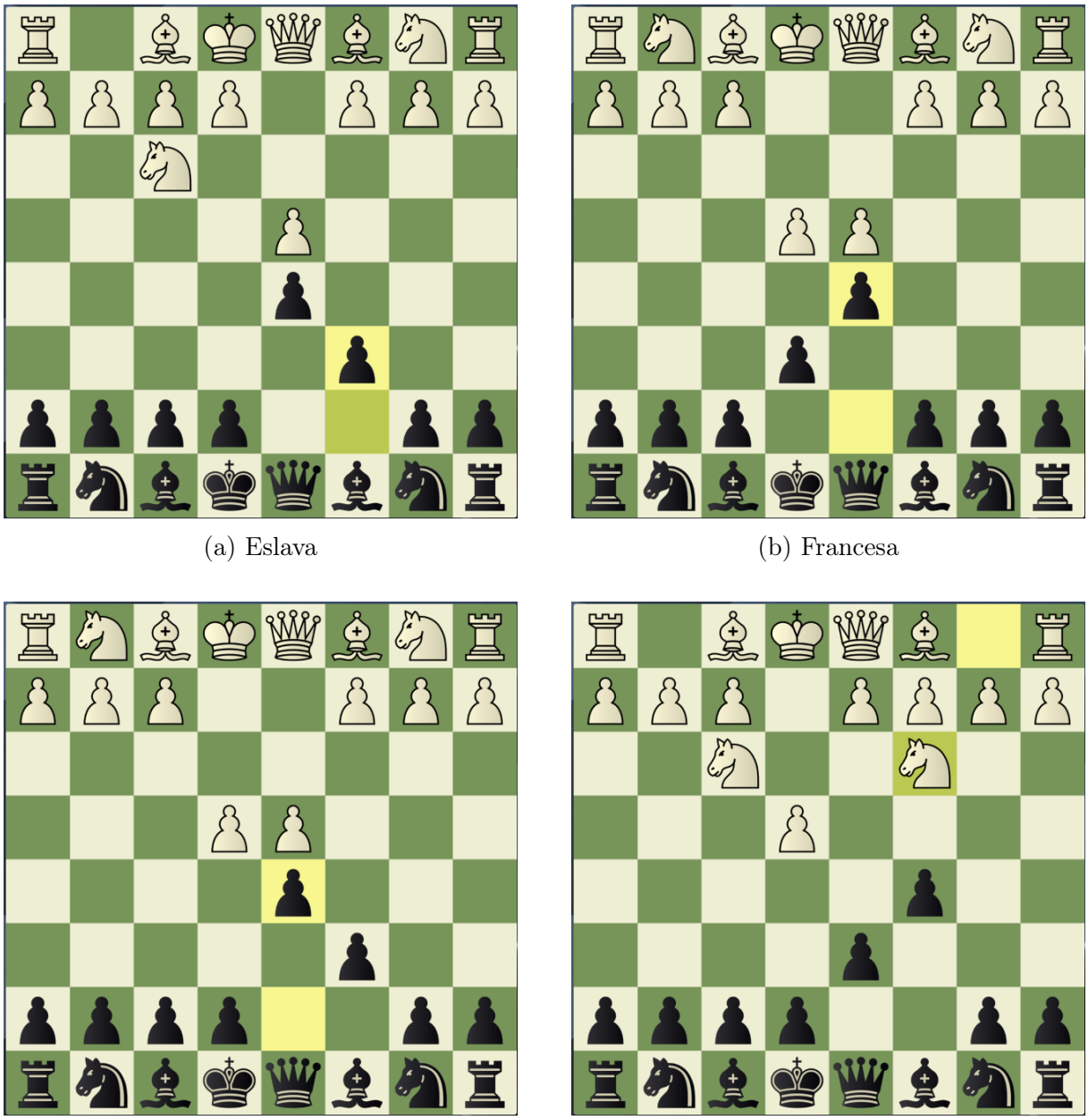
(a) Italiana/Española

(b) Inglesa

(c) Sistema Londres

(d) Fianchetto

Figura 1: Aperturas comunes en ajedrez.

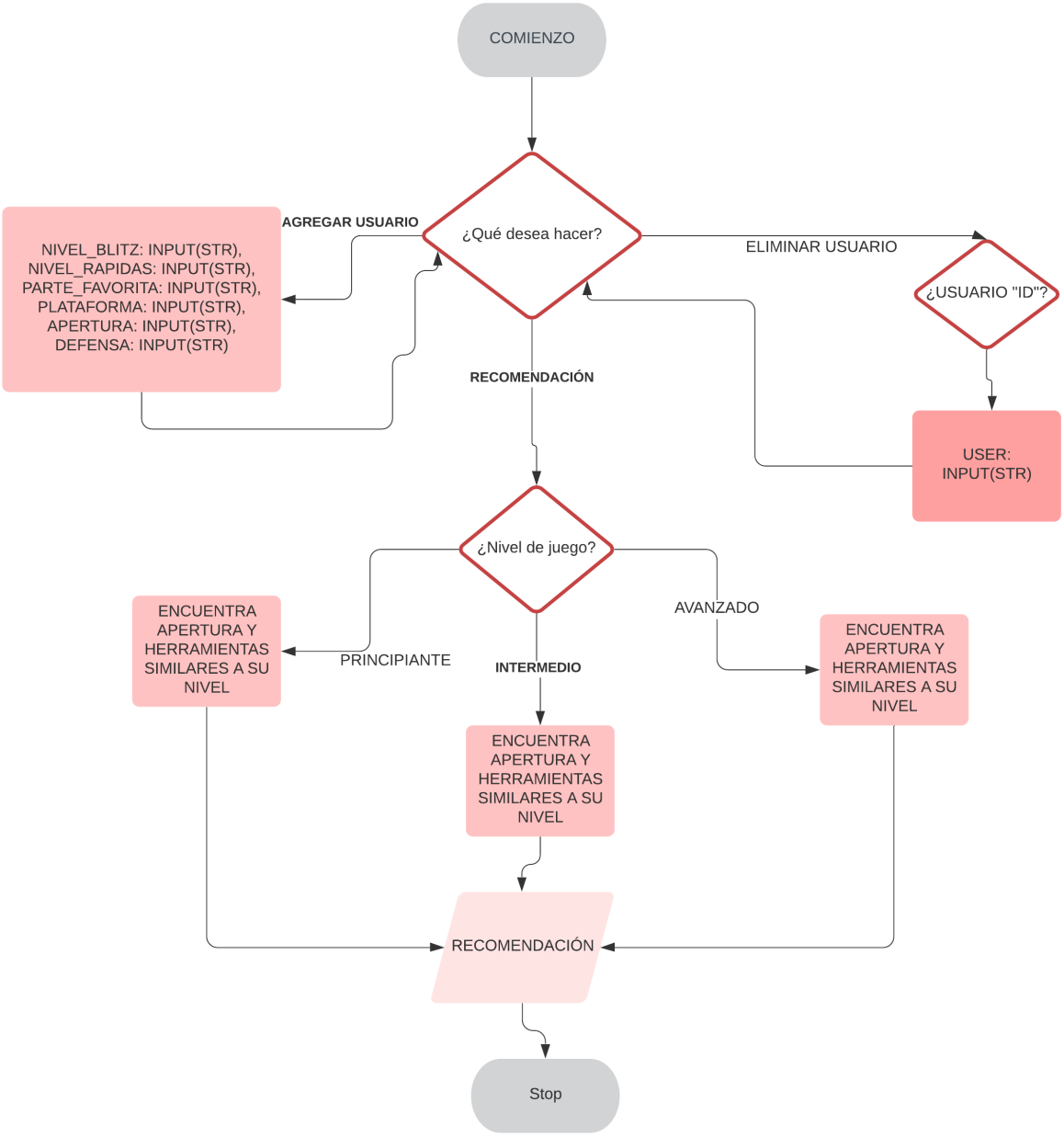


(a) Eslava (b) Francesa

(c) Caro-Kann (d) Siciliana

Figura 2: Defensas comunes en ajedrez.

3. Diagrama de flujo



4. Diseño de la base de datos

Un resumen de la base de datos original obtenida es la presentada en la parte inferior. Su implementación se basa en relacionarla con grafos. El propósito:

- **Nodos** - Los valores de las columnas.
- **Relaciones** - Los títulos de las columnas, exceptuando «USER».

USER	NIVEL_BLITZ	NIVEL_RAPIDAS	PARTE_FAVORITA	PLATAFORMA	APERTURA	DEFENSA
user001	Principiante	Intermedio	Intermedio	Chess.com	Inglesa	Siciliana
user002	Principiante	Principiante	Final	Chess.com	Fianchetto	Siciliana
user003	Principiante	Principiante	Final	Chess.com	Italiana/Española	Francesa
user004	Principiante	Principiante	Final	Chess.com	Italiana/Española	Francesa
user005	Principiante	Principiante	Intermedio	Chess.com	Italiana/Española	Francesa
user006	Principiante	Principiante	Intermedio	Chess.com	Italiana/Española	Eslava
user007	Avanzado	Avanzado	Intermedio	Chess.com	Sistema Londres	Caro-Kann
user008	Principiante	Principiante	Final	Lichess.org	Fianchetto	Caro-Kann
user009	Intermedio	Intermedio	Apertura	Chess.com	Italiana/Española	Siciliana
user010	Principiante	Principiante	Final	Chess.com	Italiana/Española	Francesa

5. Repositorio

El repositorio del proyecto se encuentra en el siguiente link:
https://github.com/RudiksChess/Grafos

Referencias

Needham, M. and Hodler, A. E. (2019). *Graph Algorithms: Practical Examples in Apache Spark and Neo4j*. O'Reilly Media.