

# Proyecto Fase #1

Carlos Calderón (15219)

Marisol Barillas(15307)

Jorge Azmitia(15202)

## Algoritmos y estructuras de datos



Universidad del Valle de Guatemala

Facultad de Ingeniería

Guatemala

2016

# Design Thinking

## **Empatía y definición**

### Tema de interés

**Recomendaciones de catedráticos:** Un sistema de recomendaciones basado en grafos. Que ayude a alumnos con sus asignaciones de catedráticos que se dan semestre a semestre.

### Perfiles

Para el sistema de recomendaciones , se especificó un solo perfil en el ámbito académico. Los cuales son las personas con mayor auge y número en la Universidad del Valle de Guatemala. En este caso, se escogió a los alumnos, pues son ellos quienes pueden presentar alguna necesidad o inquietud respecto a la manera en que se asignan catedráticos de manera semestral.

**Alumnos:** Un grupo de personas que estudian una carrera en específico dentro de la universidad. A ellos se les preguntó para los posibles usos que se le podrían dar en su contexto actual y en qué época de su vida estudiantil les hubiera sido más útil esta herramienta.

### Entrevistas

Adjuntas en otro documento por fines estéticos.

### Insights

En base a las entrevistas con los distintos perfiles de personas con las que logramos interactuar en la universidad se plantean los siguientes insights:

- Los estudiantes confían en el criterio de sus compañeros respecto a escoger catedrático.
- En la mayoría de las veces escogen a la suerte los catedráticos, siendo esto más común en los alumnos de primer año.

## Definición del Problema y Proyecto

Muchos estudiantes en la Universidad del Valle de Guatemala especialmente en primer año presentan dificultades al momento de escoger catedráticos en sus asignaciones semestrales. Este problema se presenta debido a que como es de esperar, los alumnos nuevos no tienen muchas referencias de catedráticos. De igual manera, esta situación tiene una significativa frecuencia en segundo año. Por otra parte, en los años posteriores ya la situación es un tanto distinta. Pues ya a ese punto de la carrera usualmente las opciones de asignación son más escasas, pues los cursos más específicos de la carrera son impartidos por un grupo más reducido de gente.

En base a los insights y a las entrevistas, que manifestaron los potenciales usuarios, se llevará a cabo un sistema de recomendaciones. En primera instancia, se planea un Sistema basado en Filtrado Colaborativo. Pues muchos usuarios basan su asignación en amistades cercanas, usualmente las amistades tienen rasgos similares con el usuario. Esto implica que las aptitudes que tienen son similares. Sin embargo, hay algunos casos en los que hay amistades con criterios distintos para asignar un profesor. Es por esto que se planea un Sistema basado en Filtrado Colaborativo. Ya que este consiste en ver qué usuarios son similares al usuario activo y a continuación, recomendar aquellos catedráticos que no han sido votados por el usuario activo y que han resultado bien valorados por los usuarios similares. Estas similitudes estarían basadas en ciertos rasgos característicos del perfil estudiantil del usuario.

## **Ideación**

### Reflexión Lluvia de Ideas

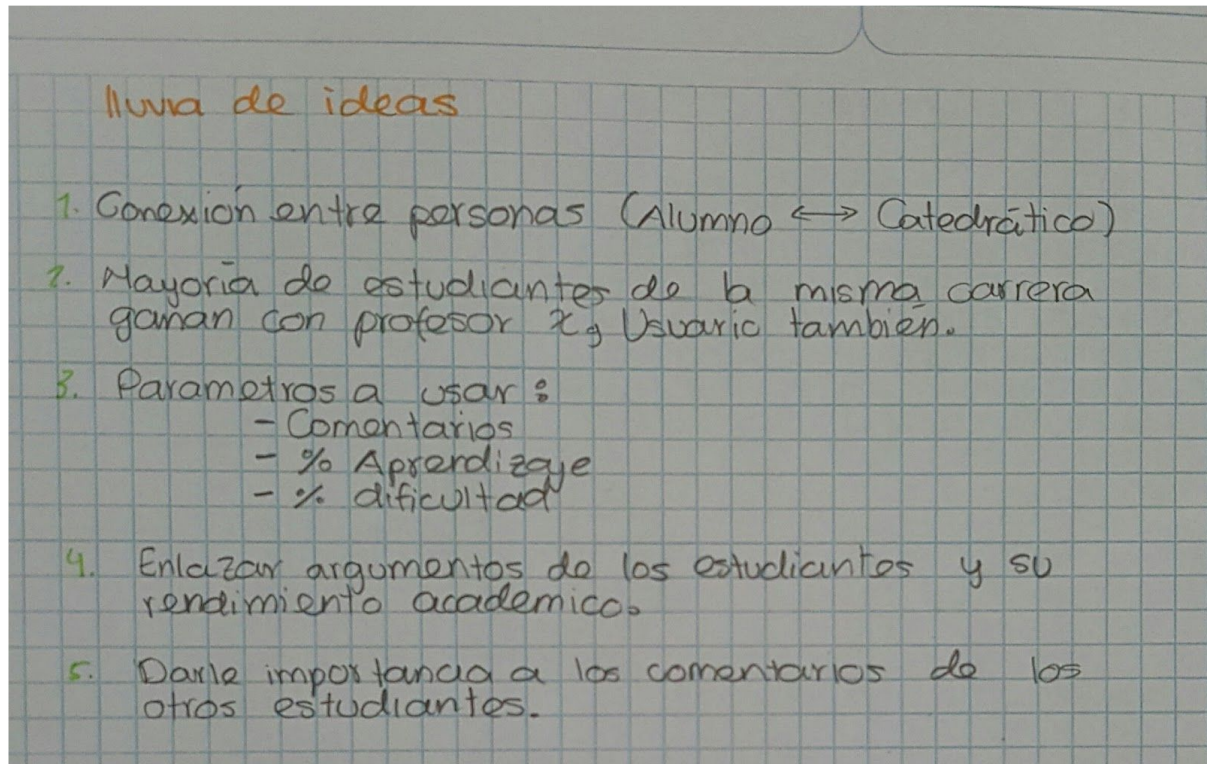
- Cada usuario tiene parámetros diferentes para escoger a un buen catedrático, por lo tanto se planea tomar en cuenta las características del usuario y compararlas con las características de los profesores y así dar la mejor referencia del mismo. Esto hace menos subjetiva la selección.
- Como bien sabemos cada persona tiene diferentes métodos para seleccionar un catedrático acorde a sus necesidades, y como vimos en las entrevistas, la mayoría se deja llevar por la experiencia de sus compañeros. Por lo tanto, se planea tener diferentes parámetros de los cuales el usuario podrá elegir cual es el que a él le interesa, ya sea comentarios, porcentaje de aprendizaje, dificultad del curso con dicho profesor, etc.
- Poniendo de base la segunda idea, uno de los parámetros será “dificultad del curso con dicho profesor”, pero se tomará en cuenta el rendimiento del estudiante a lo largo de su carrera. Esto se hará para que el usuario vea la veracidad de su argumento.
- También se tomará en cuenta la carrera de los estudiantes, ya que los estudiantes de la misma carrera tienden a aprender de la misma manera, y estudian también de formas similares.

### Conclusión de Lluvia de Ideas

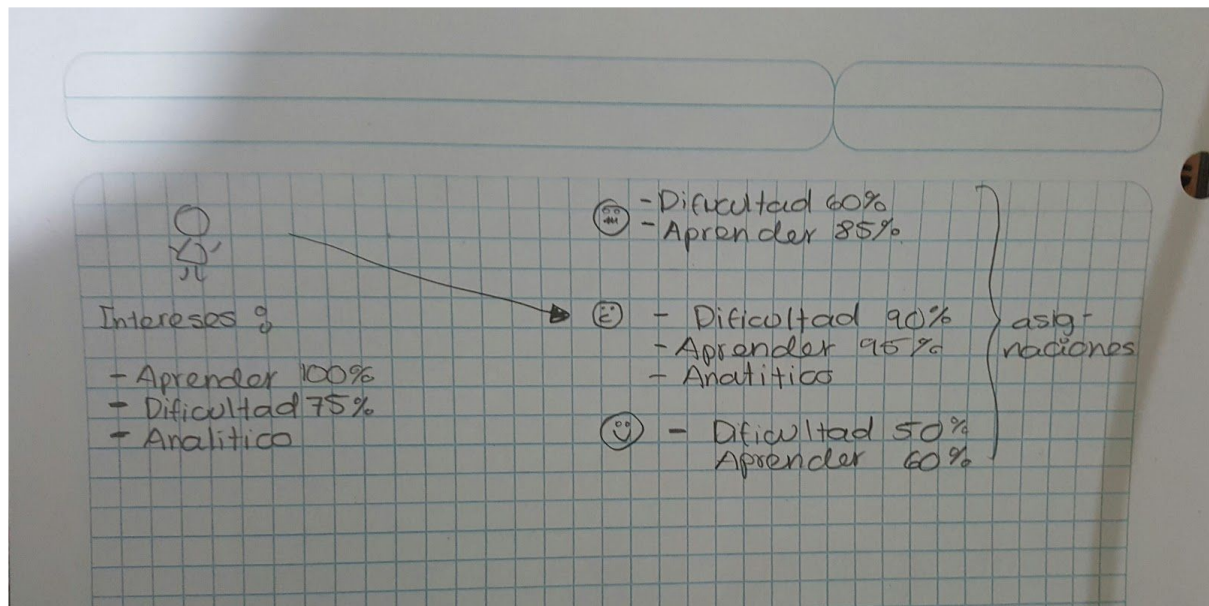
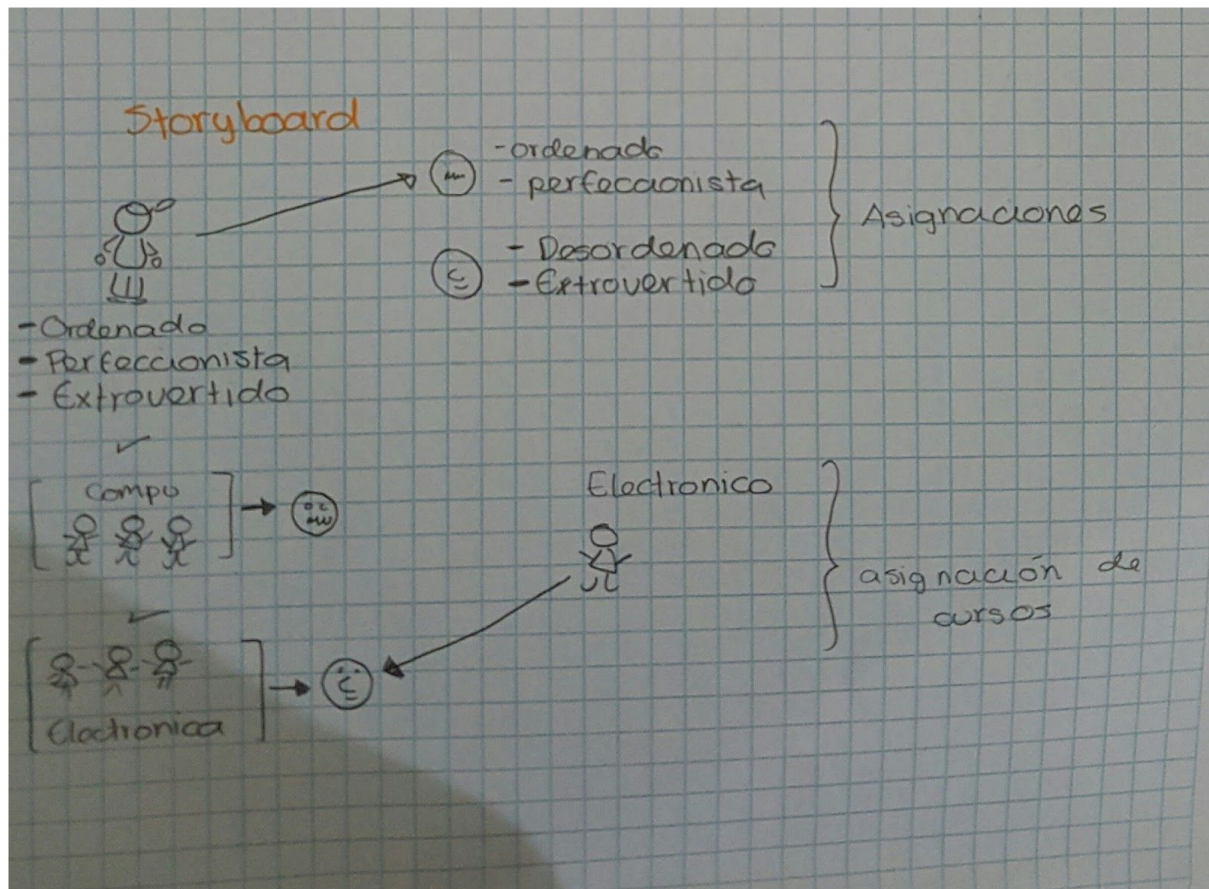
- Parámetros a utilizar:

- Porcentaje de aprendizaje.
- Dificultad del curso con el catedrático.
- Promedio del estudiante.
- Carrera del estudiante.
- Los usuarios escogen catedráticos en base a recomendaciones de amigos. Lo cual va a ser una de las principales conexiones entre el usuario y el catedrático.

## Evidencias de Lluvia de Ideas



## Storyboard



## Solución propuesta

Se propone llevar a cabo un programa con bases de datos basado en grafos. En el cual el usuario pueda tener una recomendación óptima que se ajuste a sus necesidades. El programa va dirigido principalmente a gente de primer y segundo año, pues es donde principalmente hay una amplia gamma de catedráticos en cursos de distinta índole. Asimismo, es una etapa, (principalmente



en primer año) donde recién se comienza a tomar conocimiento sobre los catedráticos en la universidad.

## Prototipar y testear

### Prototipo

A hand-drawn prototype of a form titled "DATOS:" on graph paper. The form contains several input fields and a list of interests.

**DATOS:**

Catedrático:

Curso:

Intereses:

- Organización
- Analítica
- Despreocupado
- Claridad en la explicación
- Explicación gráfica
- trabajar en grupo
- trabajo individual

Dificultad:

Aprendizaje:

A hand-drawn prototype of a form titled "Sugerencias" on graph paper. The form contains a text input field, a label for characteristics of interest, a table for years of experience, and navigation buttons.

**Sugerencias**

Nombre:

Características de interés:

Años de experiencia:	<input type="text"/>
-	-
-	-
-	-
-	-

Investigación de algoritmos existentes

Dentro de los sistemas de recomendaciones, existen diferentes técnicas de cómo filtrar la información almacenada para obtener las recomendaciones deseadas. Una de ellas es la técnica del filtrado colaborativo, la cual se basa en las similitudes entre los usuarios o los objetos. Más concretamente, se basa en la idea de que si a dos personas les gusta el mismo objeto, es muy probable que tengan la misma opinión en el resto de objetos que les gustan a cada uno. Por ello los algoritmos investigados se basan en esta técnica.

## Nearest Neighborhood

Uno de los algoritmos más usado comúnmente en sistemas recomendadores es el Nearest Neighborhood. En una red social, un usuario particular usa un vecino con gustos e intereses similares que pueden ser encontrados calculándolos con una correlación de Pearson, colectando la información de preferencia de los primeros N-vecinos para un usuario en particular, (pesado por su similitud), la preferencia del usuario se puede predecir usando ciertas técnicas.

## Heat Spreading

El algoritmo Heat Spreading es un algoritmo que destaca, principalmente, por ofrecer recomendaciones diversas a los usuarios. Para ello, el algoritmo distribuye los pesos utilizando la matriz que se presenta seguidamente, de la cual podemos destacar su normalización por filas:

$$W_{\alpha\beta}^H = \frac{1}{k_\alpha} \sum_{j=1}^u \frac{a_{\alpha j} a_{\beta j}}{k_j}$$

Esto es equivalente a distribuir, en dos pasos, los pesos de los objetos sobre el grafo bipartito. En un primer paso, cada usuario recibe un peso igual a la media de los pesos que poseen sus vecinos (dividiendo la suma de los pesos de todos los vecinos por el grado del usuario,  $k_i$ ). En el segundo paso, se realiza el mismo procedimiento, siendo los objetos los que reciben un peso igual a la media de los pesos que poseen sus vecinos (dividiendo la suma de los pesos de todos los vecinos por el grado del objeto,  $k_\alpha$ ).

### Pseudocódigo

**Entrada:** Un grafo y un usuario.

**Salida:** conjunto de objetos

- 1: A cada vecino del usuario objetivo, asignar un peso inicial de 1.
- 2: para cada objeto  $o$  que tenga un peso diferente de 0 hacer
- 3: para cada usuario  $u$  vecino de objeto hacer
- 4: Asignar a  $u$  el peso de  $o$  dividido entre el grado de  $u$ .
- 5: fin para cada
- 6: fin para cada
- 7: para cada usuario  $u$  que tenga un peso diferente de 0 hacer

- 8: para cada objeto o vecino de usuario hacer
- 9: Asignar a o el peso de u dividido entre el grado de o.
- 10: fin para cada
- 11: fin para cada
- 12: Ordenar los objetos decrecientemente según su peso final.

## Probability Spreading

El algoritmo Probability Spreading es un algoritmo que destaca, principalmente, por ofrecer recomendaciones precisas a los usuarios. Para ello, el algoritmo distribuye los pesos utilizando la matriz que se presenta seguidamente, de la cual podemos destacar su normalización por columnas:

$$W_{\alpha\beta}^p = \frac{1}{k_\beta} \sum_{j=1}^u \frac{a_{\alpha j} a_{\beta j}}{k_j}$$

Esto es equivalente a distribuir, en dos pasos, los pesos de los objetos sobre el grafo bipartito. En un primer paso, para cada objeto se distribuye su peso repartido equitativamente entre todos sus usuarios vecinos (dividiendo el peso por el grado del objeto,  $k_\beta$ ). En el segundo paso, los usuarios que han recibido los pesos los redistribuyen a sus objetos vecinos, repartiéndolos equitativamente también (dividiendo el peso por el grado del usuario,  $k_j$ ).

### Pseudocódigo

**Entrada:** Un grafo y un usuario.

**Salida:** Conjunto de objetos recomendados, en orden descendiente.

- 1: A cada vecino del usuario objetivo, asignar un peso inicial de 1.
- 2: para cada objeto o que tenga un peso diferente de 0 hacer
- 3: para cada usuario u vecino de objeto hacer
- 4: Asignar a u el peso de o dividido entre el número de vecinos de o.
- 5: fin para cada
- 6: fin para cada
- 7: para cada usuario u que tenga un peso diferente de 0 hacer
- 8: para cada objeto o vecino de usuario hacer
- 9: Asignar a o el peso de u dividido entre el número de vecinos de u.
- 10: fin para cada
- 11: fin para cada
- 12: Ordenar los objetos decrecientemente según su peso final.

## Algoritmo híbrido



El algoritmo Probability Spreading ofrece un gran precisión a la hora de realizar las recomendaciones en detrimento de una baja diversidad; mientras que, el algoritmo Heat Spreading ofrece unas recomendaciones con un grado de diversidad mucho mayor pero penalizando la precisión. Por lo tanto, con el objetivo de lograr un algoritmo que obtenga tanto recomendaciones precisas como diversas, se ha construido un algoritmo híbrido a partir de los dos anteriores para intentar beneficiarse de las características de ambos. Dicho algoritmo distribuye los pesos utilizando la matriz que se presenta seguidamente:

$$W_{\alpha\beta}^{H+P} = \frac{1}{k_{\alpha}^{1-\lambda} k_{\beta}^{\lambda}} \sum_{j=1}^u \frac{a_{\alpha j} a_{\beta j}}{k_j}$$

Esto es equivalente a distribuir, en dos pasos, los pesos de los objetos sobre el grafo bipartito. Más concretamente, se trata de realizar los mismos pasos que en los anteriores algoritmos pero simultáneamente, otorgando más poder a un algoritmo o otro según el factor de lambda. De esta manera, se puede afinar  $\lambda$  para aumentar las características de un algoritmo o otro, según convenga o nos interese.

#### Pseudocódigo

**Entrada:** Un grafo, un usuario y un número decimal entre 0 y 1 ( $\lambda$ ).

**Salida:** conjunto de objetos

- 1: A cada vecino del usuario objetivo, asignar un peso inicial de 1.
- 2: para cada objeto o que tenga un peso diferente de 0 hacer
- 3: para cada usuario u vecino de objeto hacer
- 4: grado := grado del usuario u
- 5: vecinos := número de vecinos del objeto o
- 6: tmp :=  $\text{grado}^{(1-\lambda)} * \text{vecinos}^{\lambda}$
- 7: Asignar a u el peso de o dividido entre tmp.
- 8: fin para cada
- 9: fin para cada
- 10: para cada usuario u que tenga un peso diferente de 0 hacer
- 11: para cada objeto o vecino de usuario hacer
- 12: grado := grado del objeto o
- 13: vecinos := número de vecinos del usuario u
- 14: tmp :=  $\text{grado}^{(1-\lambda)} * \text{vecinos}^{\lambda}$
- 15: Asignar a o el peso de u dividido entre tmp.
- 16: fin para cada
- 17: fin para cada
- 18: Ordenar los objetos decrecientemente según su peso final.