

## **Fase 1: Identificación del Problema:**

### **Descripción del Contexto Problemático:**

Para el curso de Modelado de Sistemas de Información se asignó la realización de una agenda de contactos en la que se encuentre la información personal, de contacto y sobre los clases matriculadas de todos los estudiantes del curso. Sin embargo, la propuesta del curso es que esta agenda sea capaz de generar sus contactos a partir de una base de datos en texto plano.

Inicialmente se dieron estas indicaciones, con ausencia de detalles, por lo que los distintos grupos de trabajo encontraron difícil el identificar la información que debería tener cada una de las entradas en la agenda. Diversas propuestas se hicieron, inicialmente creando propuestas de interfaces de usuario que constaban de diferentes campos de información con datos sobre los estudiantes y sus clases actuales.

Es desde este enfoque que surge la necesidad de el curso de Modelado de Sistemas de Información, pues un grupo de desarrollo ha de ser firme en el tipo y forma de la información a ser usada, y los datos que serán relevantes para un sistema capaz de funcionar correctamente.

### **Identificación de Necesidades:**

Las siguientes fueron las necesidades identificadas en el programa a ser desarrollado:

- El programa debe de ser una agenda de contactos con información sobre los cursos que llevan estos.
- La agenda deberá llenar su información a partir de una base de datos preestablecida.
- Sólo los datos relevantes a la agenda serán usados en esta.
- La agenda deberá ser capaz de genera reportes a partir de la información en esta. Estos reportes incluyen:
  - Promedio de materias matriculadas por el estudiante.
  - Promedio de créditos que un estudiante tiene matriculados.
  - Materia más matriculada, incluyendo el número de estudiantes matriculados.
  - Materia menos matriculada, incluyendo el número de estudiantes matriculados.
  - Un listado con todas las materias matriculadas por los estudiantes, permitiendo identificar la materia más y menos matriculadas en la agenda.
- La agenda deberá almacenar la información registrada de los alumnos.
- La agenda permitirá agregar nuevos registros.
- Las modificaciones realizadas en la agenda deberán actualizarse también en la base de datos.

## **Fase 2: Recopilación de la información:**

### **Marco teórico**

Los siguientes conceptos son necesarios para la comprensión de lo explicado en este documento.

- Base de Datos: Son conjuntos de datos reunidos y almacenados, interesantes para un contexto en específico. Usualmente contienen entradas y campos, en donde cada

entrada corresponde a una unidad de datos reunida, y cada campo se refiere a la información relevante de dicha entrada.

- CSV: Son archivos de texto planos que representan tablas. Cada columna de la tabla es separada por una coma. Son útiles para almacenar información en un formato ligero. Posee restricciones dado que su formato no es estandarizado y porque presenta problemas cuando la información a almacenar contiene comas.
- Agendas de contactos: Dan la opción de almacenar diversos tipos de información de un contacto según las necesidades del usuario. Permiten buscar por el nombre de la persona de una entrada en específico.

### Estado del arte:

A continuación se exponen diversas soluciones y herramientas útiles para este problema:

- Sheets de Google: La aplicación en línea para manejo de hojas de cálculo de Google. Como cualquier aplicación del conjunto de aplicaciones para oficina de Google, esta permite que cualquier documento pueda ser abierto a un determinado grupo de personas. Al crearse una hoja de cálculo libre para edición por los estudiantes del curso, se puede crear una lista con la información relevante de los contactos que serán usados en la agenda, permitiendo su uso como una base de datos. Bastará con exportar el archivo, una vez editado, como un archivo de texto fácilmente procesable por la aplicación desarrollada.
- BufferedReader de Oracle: La librería JavaIO de Oracle, implementada en Java, contiene la clase BufferedReader, la cual es capaz de procesar archivos de texto en el sistema de archivos de un programa. Dicha clase, dada una ruta con la ubicación de un archivo, permite que la información en este pueda ser manejada por un programa escrito en Java. Esta clase, y esta librería en general, pueden ser usadas para leer la base de datos ya creada con la información de contacto, siempre que esta sea tratada como un archivo de texto con separadores explícitos.
- Dictionary de Oracle: La librería Collections de Java, por Oracle, incluye la estructura de datos Dictionary, una que almacena conjuntos de datos por pares clave-valor. Si se utiliza este tipo de estructura de datos, se pueden almacenar los contactos de la agenda por claves únicas para estos, como lo es su código estudiantil, lo que otorgaría la ventaja de realizar búsquedas en tiempos de cómputo extremadamente eficientes. Esto conlleva a una aplicación bastante rápida y eficaz.

### **Fase 3: Búsqueda de Soluciones Creativas:**

Para esta fase se tuvo en cuenta que las problemáticas principales son dos: modelar y diseñar un conjunto de datos que mantenga la información relevante para una aplicación como lo es una agenda, y poder procesar ese conjunto de datos como una base de datos.

Para el diseño del conjunto de datos relevante se planteó inicialmente que cada grupo de trabajo pensara en los datos que considerase relevante para la aplicación, lo que llevó a la problemática expuesta inicialmente en la fase 1. Debido a esto, se llegó a un consenso: crear una hoja de cálculo que cualquier estudiante del curso de Modelado de Sistemas pudiera editar, y que esta se llenara con los datos de cada uno. Sin embargo, como todavía no se había definido el tipo de información que se desearía guardar de cada uno de los

estudiantes, se optó por realizar una discusión grupal, en la que cada los estudiantes fueron aportando sus ideas respecto a la información necesaria para esta labor. Desde cierto punto de vista, se puede pensar que este problema fue abordado mediante la técnica de Brainstorming. Una vez se llegó al consenso de la información a mantenerse, cada estudiante fue a llenar la hoja de contactos creada, y se planea usar dicha hoja de contactos cómo la base de datos de la aplicación. Ahora bien, como cada grupo planea mantener sus propias ideas respecto a la información deseada, se piensa utilizar esta hoja de contactos como la información básica, y que cada grupo implemente la información que haga falta como datos en campos adicionales.

Por otro lado, respecto al manejo de esta información, se planea utilizar una estructura de datos que facilite la edición de entradas individuales en la base de datos, así como búsquedas bajo ciertos criterios en esta. Para esto, se proponen las alternativas listadas a continuación:

- Dictionary. Esta estructura de datos almacena la información mediante pares clave-valor, que, dado una clave única en el diccionario, retornan el valor asociado a dicha clave.
- Array. El arreglo es una estructura de datos que almacena sus datos mediante índices numéricos. Se puede pensar un arreglo como un conjunto de casillas en las que se guardará información, y a la que se puede acceder conociendo el número de cada casilla.
- List. Es una estructura de datos bastante similar a el arreglo, con la diferencia que las listas permiten el aumento y disminución de su tamaño, lo cual le otorga una ventaja sobre los arreglos. A cada entrada en una lista se le asigna un índice único para esta, y conocido el índice se puede acceder a los datos de este.

#### **Fase 4: Transición de la Formulación de Ideas a Diseños Preliminares:**

Como se explicó en la fase anterior, la problemática respecto a la información que se desea almacenar y la manera en que esta sería recogida ya fueron abordados y resueltos, por lo que solo queda por resolver lo concerniente al manejo de esta información.

- Alternativa 1: Dictionary  
Bajo el concepto de diccionario, se puede pensar en cada clave como un dato único de cada estudiante, como lo puede ser su código o su número de identificación; y el valor asociado a dicha clave sería toda la demás información almacenada de este estudiante. El problema en el uso de diccionarios radica en lo complejo que resulta realizar operaciones de ordenamiento sobre estos, pues no mantienen un orden fijo en cómo guardan sus entradas.
- Alternativa 2: Array  
Si se utiliza esta estructura de datos, se puede utilizar cada una de las entradas en el arreglo como una entrada de la agenda; y, dado el índice que le corresponda, se puede acceder a su información. Sin embargo, debido a su tamaño fijo y limitado, no permitirían el agregar nuevas entradas a la agenda, por lo que son descartados como una solución a este problema.
- Alternativa 3: List  
Las listas, de manera similar a los arreglos, se pueden utilizar para guardar cada una

de las entradas de la agenda como una entrada de la lista. Por su tamaño flexible, son ideales para operaciones de adición y sustracción de entradas en la agenda, además de permitir realizar algoritmos de ordenamiento sencillos sobre estas. El problema con este tipo de estructura de datos es que, al guardar los datos mediante índices y sin utilizar ningún orden en específico, las operaciones de búsqueda resultan algo más lentas que en otras estructuras.

### **Fase 5: Evaluación y Selección de la Mejor Solución:**

Para esta fase se definieron los siguientes criterios para evaluar las soluciones propuestas:

- Accesibilidad de las entradas: En este criterio se califica la capacidad de acceso de la estructura de datos escogida para acceder con facilidad la información almacenada en esta, donde una calificación de 0 indica que la estructura no permite acceder a la mayoría de los datos, una calificación entre 0 y 3 indica que la estructura accede con dificultad (tiempo mayor a  $O(n)$ ) a los datos, y 3 indica que la estructura accede sin dificultad a los datos almacenados en esta.
- Capacidad de Edición de la Base de Datos: En este criterio se califica la capacidad de un programa para modificar los datos almacenados en la base de datos. Una calificación de 0 indica que no se pueden modificar la base de datos, y una calificación de 3 indica que se pueden realizar modificaciones por parte del usuario sobre la base de datos, bajo ciertas restricciones.
- Facilidad de Implementación: En este criterio se mide la facilidad con la que las necesidades expuestas para el programa puedan ser implementadas, donde 0 indica que algunas necesidades no pueden ser implementadas en lo absoluto, una calificación entre 0 y 3 indican que pueden ser implementadas, pero bajo procesos complejos y poco eficaces, y un 3 indica que todas las necesidades pueden ser resueltas sin mayor inconveniencia para los desarrolladores.

En la siguiente tabla se resumen los resultados de este proceso de calificación:

Alternativa	Accesibilidad de las Entradas (40%)	Capacidad de Edición de la BD (40%)	Facilidad de Implementación (20%)	Total (100%)
Dictionary	2.7	3	2	2.68
List	3	3	2.5	2.9

Cómo se observa, la estructura List es la que mejor cumple con los criterios explicados anteriormente, por lo que será la utilizada para el desarrollo de la aplicación.

Sin embargo, una tarea como lo es contar frecuencias en un grupo de datos, necesaria para obtener las materias más y menos matriculadas del grupo, es bastante tediosa de realizarse en una estructura de datos como List. Sin embargo, esta tarea se realiza de manera sencilla al utilizar estructuras de tipo diccionario, por lo que se utilizará esta estructura de manera parcial para resolver esa necesidad en específico.