Modelado de sistemas de información

Proyecto 1 - 2020-1

César Canales - A00345026, Christian Flor - A00355624 Daniel Fernández - A00354694 Carlos Restrepo - A00355028

Contexto problemático

Las agendas digitales son una manera de organizar nuestra información de manera eficiente y compacta, la cual facilita la utilización de esta, para esta asignación se considera la implementación de una agenda digital para los estudiantes del curso de “Modelado de sistemas de información”.

Fase 1. Identificación del problema

Identificación de los síntomas y necesidades de la situación problemática

- Se requiere una manera de manipular la información de los estudiantes, esto es, poder cargarla, modificarla y guardarla.
- Es necesario que el usuario pueda visualizar información acerca de la materia más matriculada, la menos matriculada, etc.
- El usuario debe ser capaz de agregar y quitar estudiantes o materias a su agenda digital.
- La aplicación debe funcionar como una agenda digital, esto es, cualquiera puede modificar la información contenida en esta a su gusto.

Definición del problema

No existe una manera sencilla de manipular la información de los estudiantes del curso de Modelado de sistemas de información.

Fase 2. Recopilación de información

Agenda digital

La agenda digital se define formalmente como un dispositivo o programa que permite almacenar apuntes o información de importancia para su posterior uso.

Persistencia de datos

La persistencia de datos es la representación residual de datos que han sido de alguna manera nominalmente borrados o eliminados. Este residuo puede ser debido a que los

datos han sido dejados intactos por un operativo de eliminación nominal, o por las propiedades físicas del medio de almacenaje. La persistencia de datos posibilita en forma inadvertida la exhibición de información sensible si el medio de almacenaje es dejado en un ambiente sobre el que no se tiene control (p. ej., se tira a la basura, se le da a un tercero).

JSON

JSON (JavaScript Object Notation) es un formato ligero de intercambio de datos. Es fácil para los humanos leer y escribir. Es fácil para las máquinas analizar y generar. Se basa en un subconjunto del lenguaje de programación JavaScript Standard ECMA-262 3rd Edition - December 1999. JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son familiares para los programadores de la familia de lenguajes C, incluido C, C ++, C #, Java, JavaScript, Perl, Python y muchos otros. Estas propiedades hacen de JSON un lenguaje ideal para el intercambio de datos.

Estado del arte

Hay incontables aplicaciones de agendas digitales que permiten hacer una variedad de cosas, no solamente mantener la información de un curso, por ejemplo, Academity es una aplicación que permite el manejo de una completa agenda de clase que te permite realizar: control de asistencia, estadísticas del alumno, resúmenes de clase, asignación de tareas grupales o individuales.

Fase 3. Búsqueda de soluciones creativas

La manipulación de la información tiene dos subproblemas los cuales son, la persistencia de la información y el grado en que el usuario puede modificar la información. La persistencia es la manera en cómo se va a guardar la información(archivo de texto, servidor web, etc), lo segundo es definir qué campos podrá el usuario cambiar y en qué momento lo puede hacer. Sin embargo, estos dos problemas se pueden juntar con el supuesto de que en el momento que se modifique la información, la persistencia debe actuar para hacer que la información que fue modificada deje de persistir.

Teniendo lo dicho hasta aquí en cuenta, se implementará la idea una agenda digital en la cual toda la información es modificable, es decir, no hay límite alguno para modificar los distintos campos de los estudiantes.

Por tanto, se consideran un problema y se utiliza la técnica de lluvia de ideas para la generación de ideas. Utilizando como pilar principal las distintas formas de hacer persistencia de la información.

★ **Ideas propuestas:**

Alternativa 1. Archivo de texto plano
Esta idea consiste en guardar la información de los estudiantes y de las materias en dos archivos de texto distintos, esto permite que en el momento que se cargue la información simplemente se tengan que leer dos archivos y, así mismo, al momento en que se cierre la aplicación simplemente se sobreescriben estos dos archivos con los nuevos objetos.

Alternativa 2. Servidor web
Esta alternativa consiste en la utilización de un servidor web donde se guarde toda la información de los estudiantes y las materias, esto brinda más seguridad ya que los datos quedan en la nube. Por tanto, siempre que el usuario tenga conexión a internet este podrá acceder a la información.

Alternativa 3. Utilizar archivos tipo properties
Este enfoque consiste en tener distintos archivos de propiedades donde cada uno representa un estudiante o una materia, esto facilita las modificaciones de la información por estudiante y además las funciones de eliminación y adición de estudiantes ya que es solo borrar o crear un archivo respectivamente.

Fase 4. Transición de las ideas a los diseños preliminares

Primeramente, procedemos a descartar ideas inalcanzables.

Alternativa 2. Servidor web
Descartamos esta alternativa dado que esta requiere de un servidor web, el cual no se tiene y tampoco se tienen los recursos y el tiempo para obtener uno.

A continuación, exploramos el resto de las alternativas a más detalle

Alternativa 1.Archivo de texto plano
Esta alternativa hace todos los procesos de modificación y actualización internamente, con los objetos creados en el momento en que se inició la aplicación y en el momento en que se cierra la aplicación, con los objetos que fueron modificados internamente se sobreescribe los archivos de texto y así la información queda guardada.

Alternativa 3. Utilizar archivos tipo properties

Esta alternativa a diferencia de la anterior realiza los procesos de modificación y actualización no solo internamente sino también externamente, esto es, cuando se modifica algún dato, se modifica en la aplicación su respectivo objeto, pero además se modifica su respectivo archivo de texto de manera simultánea.

Fase 5. Evaluación y selección de la mejor solución

Los criterios que van a permitir la evaluación de las alternativas propuestas para así seleccionar la solución que mejor cumple con los requerimientos se lista a continuación

- Facilidad de la implementación: Este criterio se evalúa de 0 a 5, el 0 quiere decir que la implementación de la solución tiene una complejidad bastante alta donde se requiere el uso de algoritmos no triviales o de muchas líneas de código y el 5 significa que la solución es fácil y corto de implementar y no requiere de muchos algoritmos complejos para lograr su funcionamiento.
- Propenso a errores: Este criterio se evalúa de 0 a 3, el 0 quiere decir que la alternativa no es muy propensa a tener errores no deseados a la hora de su funcionamiento y el 3 significa que la idea es muy propensa a causar errores en el programa y hacer que este no funcione de la manera adecuada. Este criterio no suma sino que resta en el total de la evaluación.
- Eficiencia de la solución: Este criterio se evalúa de 0 a 5, el 0 quiere decir que se cumple la función de persistencia más no de la manera más efectiva posible, esto puede ser porque no tiene la mejor complejidad temporal, porque realiza procesos innecesarios, etc. El 5 quiere decir que se cumple la función de persistencia de una manera muy efectiva.

Evaluación

Utilizando los criterios mencionados anteriormente, obtenemos la siguiente tabla al evaluar las alternativas

Persistencia de la información				
	Facilidad de la implementación	**Propenso a errores**	**Eficiencia de la solución**	**Total**
Alternativa 1.Archivo de texto plano	5	2	4	7
Alternativa 3. Utilizar archivos tipo properties	4	0	5	9

Selección

De acuerdo a las evaluaciones previas, se obtiene que la alternativa de utilizar archivos tipo properties es la más viable para el problema planteado. De esta manera, se tendrá un archivo por estudiantes con sus respectivos atributos, nombre, apellido, correo, código, número telefónico, programa, foto, semestre y materias. Además, se tendrá un archivo por materia con los atributos nombre, nrc, facultad, créditos y número de estudiantes cursando la materia.

Fase 6. Preparación de reportes y especificaciones

1. **Especificación de requerimientos funcionales. (en términos de entradas y salidas)**

Nombre	**F.R# 1. Crear entrada en la agenda**
Resumen	El programa permitirá registrar nuevos usuarios en la agenda. Para esto se debe proporcionar la información básica y de contacto del usuario a agregar. No se permitirá el registro de usuarios que ya lo estén.
Entradas	
Código estudiantil, nombre, apellido, correo, programa académico y semestre.	

Salidas	
Se crea una nueva entrada en la agenda si el código estudiantil ingresado ya no está registrado en la agenda.	

Nombre	**F.R# 2. Agregar materia**
Resumen	El programa permitirá al usuario inscribirse a una materia siempre y cuando no haya excedido el límite de créditos matriculables.
Entradas	
Código estudiantil del estudiante al que se le va a agregar la materia y código(nrc) de la materia.	
Salidas	
El estudiante ahora tiene una nueva materia.	

Nombre	**F.R# 3. Eliminar entrada en la agenda**
Resumen	El programa permitirá eliminar a un estudiante del sistema.
Entradas	
Código estudiantil del estudiante a eliminar.	
Salidas	
El estudiante fue eliminado con éxito si el código se encuentra en la agenda.	

Nombre	**F.R# 4. Eliminar materia**
Resumen	El programa permitirá que el estudiante elimine una materia de las que tiene matriculadas.
Entradas	
Código estudiantil del estudiante al que se le va a agregar la materia y código(nrc) de la materia.	
Salidas	
La materia fue eliminada con éxito.	

Nombre	**F.R# 5. Buscar estudiantes por nombre.**
Resumen	El programa permitirá buscar estudiantes por su nombre.
Entradas	• **Un String con el nombre del estudiante.**
Nombre del estudiante.	

Salidas	
Lista de los estudiantes con el nombre especificado o alerta de que no se encontró el estudiante con el nombre especificado.	

Nombre	**F.R# 6. Buscar estudiantes por apellido.**
Resumen	El programa permitirá buscar estudiantes por su apellido.
Entradas	
Apellido del estudiante.	
Salidas	
Lista de los estudiantes con el apellido especificado o alerta de que no se encontró el estudiante con el apellido especificado.	

Nombre	**F.R# 7. Buscar estudiantes por código**
Resumen	El programa permitirá buscar estudiantes por su código estudiantil.
Entradas	• **Un String con el código del estudiante.**
Código del estudiante.	
Salidas	• **Un boolean: verdadero si se encontró el estudiante, falso de lo contrario.** • **Información personal del estudiante.** • **Listado de materias matriculadas.**
Lista de los estudiantes con el código especificado o alerta de que no se encontró el estudiante con el código especificado.	

Nombre	**F.R# 8. Buscar estudiantes por correo electrónico.**
Resumen	El programa permitirá buscar estudiantes por su correo electrónico.
Entradas	
Correo del estudiante.	
Salidas	
Lista de los estudiantes con el correo especificado o alerta de que no se encontró el estudiante con el correo especificado.	

Nombre	**F.R# 9. Mostrar promedio de materias.**

Resumen	El programa permitirá calcular y mostrar el promedio de materias matriculadas por estudiante.
Entradas	
Ninguna	
Salidas	
Valor del promedio de materias matriculadas	

Nombre	**F.R# 10. Mostrar promedio de créditos.**
Resumen	El programa permitirá calcular y mostrar el promedio de créditos matriculados por estudiante.
Entradas	
Ninguna	
Salidas	
Valor del promedio de créditos matriculados por estudiante.	

Nombre	**F.R# 11. Mostrar materia más matriculada.**
Resumen	El programa permitirá mostrar información de la materia más matriculada por los estudiantes.
Entradas	
Ninguna	
Salidas	
Nombre, créditos, estudiantes inscritos y nrc de la materia más matriculada.	

Nombre	**F.R# 12. Mostrar materia menos matriculada.**
Resumen	El programa permitirá mostrar información de la materia menos matriculada por los estudiantes.
Entradas	
Ninguna	
Salidas	
Nombre, créditos, estudiantes inscritos y nrc de la materia menos matriculada.	

Nombre	**F.R# 13. Actualizar información de un estudiante**

Resumen	Se permitirá editar y guardar campos de información de un estudiante registrado.
Entradas	
Código estudiantil, nombre, apellido, correo o programa que representa el campo a ser modificado.	
Salidas	
Información del estudiante modificada.	

Nombre	**F.R# 14. Mostrar listado de materias.**
Resumen	El programa permitirá mostrar información de todas las materias matriculadas por los estudiantes.
Entradas	
Ninguna	
Salidas	
Un listado de las materias ordenadas alfabéticamente por su nombre.	

2. **Diagrama de clases**
3. **Diseño de la persistencia**

Síntesis reflexiva

Se utilizó el método de la ingeniería con el fin de llegar a una solución óptima, las primeras fases consisten en la identificación del problema y de la recopilación de la información necesaria para adquirir un mejor entendimiento de este. Luego, se idearon posibles soluciones al problema y se evaluaron para terminar con la mejor de ellas. Esta estrategia logró brindarle solución a todos los requerimientos propuestos. En general, se logró cumplir con los requerimientos y además, se aprendieron los conceptos básicos de las bases de datos.

Fase 7. Implementación

La solución al problema se encuentra en el repositorio:
https://github.com/ChristianFlor/digital-agenda#overview

Referencias

[1] “PDA”, Wikipedia, 2013. [En línea]. Disponible en: https://es.wikipedia.org/wiki/PDA [Accedido: 09-feb-2020]

[2] “Persistencia de datos”, Wikipedia, 2010. [En línea]. Disponible en: https://es.wikipedia.org/wiki/Persistencia_de_datos [Accedido: 09-feb-2020]
https://www.json.org/json-en.html

[3] “Persistencia de datos”, Uniwebsidad, 2011. [En línea]. Disponible en: https://uniwebsidad.com/libros/algoritmos-python/capitulo-11/persistencia-de-datos
[Accedido: 09-feb-2020]

[4] “Introducing JSON”, JSON,. [En línea]. Disponible en: https://www.json.org/json-en.html [Accedido: 09-feb-2020]

[5] “Academity”, Academity, 2020. [En línea]. Disponible en: https://academity.es/funcionalidades/ [Accedido: 09-feb-2020]