

Manejo de la vida adulta

Sofia Vargas, Jorge Alandete, Jeronimo Bermudez, Nicolas Quezada

No. de equipo de trabajo: E

I. INTRODUCCIÓN

En el presente documento se aborda el tema del manejo de la vida adulta, un período de transición que enfrentan los jóvenes al adentrarse en esta nueva etapa. Se destaca la complejidad de este proceso, el cual ha sido objeto de numerosos estudios que resaltan la diversidad de experiencias y desafíos que lo caracterizan [1]. Entre estos desafíos se encuentra la adquisición de nuevas responsabilidades sin el conocimiento o las herramientas necesarias para abordarlas adecuadamente, lo que puede tener consecuencias perjudiciales a largo plazo, tales como dificultades en la vida laboral, problemas de salud mental y dificultades en las relaciones interpersonales [5].

Para enfrentar este desafío, se adopta un enfoque multidisciplinario que resalta la importancia de adquirir habilidades y conocimientos adecuados durante este período [5]. Sin embargo, se reconoce la imposibilidad de abordar todas las variables influyentes en la transición a la adultez, por lo que se priorizarán aquellos aspectos considerados más relevantes y viables de investigar.

El presente documento también describe las funcionalidades del software desarrollado para abordar este problema, así como las estructuras de datos implementadas, los entornos de desarrollo y operación, roles y actividades del equipo, dificultades encontradas y lecciones aprendidas durante el proceso de desarrollo.

II. DESCRIPCIÓN DEL PROBLEMA A RESOLVER

La transición a una nueva etapa, en este caso la adulta es un proceso complejo que ha sido objeto de numerosos estudios, se destaca la diversidad de experiencias y desafíos que enfrentan los jóvenes durante esta transición[1], uno de los principales, el desconocimiento y la falta de herramientas necesarias para llevar a cabo las nuevas responsabilidades adquiridas, las cuales pueden tener consecuencias perjudiciales a largo plazo de no ser abordadas correctamente, estas pueden incluir dificultades en la vida laboral, problemas de salud mental y malas relaciones interpersonales[5].

Para referirnos a un contexto más local Podemos enfocarnos en Colombia, en el que se suman las diferencias socioeconómicas y cómo algunas de ellas complican esta transición [2], o podemos observar como afecta a los Jóvenes desde un panorama histórico en América Latina y las diferencias entre cada década[3].

Para abordar este problema, se empleará un enfoque multidisciplinario en el cual se destaca la importancia de adquirir habilidades y conocimientos adecuados durante este período [5], sin embargo se reconoce que este proyecto no puede abordar todas las posibles variables influyentes en la transición a la adultez, y se priorizará aquellos aspectos que se consideren más relevantes y viables de investigar.

III. USUARIOS DEL PRODUCTO DE SOFTWARE

- Grupo de funcionalidades utilizadas:

Usuarios sin cuenta: Aquellos usuarios que no se encuentran registrados en el aplicativo, solo podrán acceder a la visualización de consejos.

Usuarios básicos: Aquellos usuarios registrados que solo necesitan acceder a las funciones principales y fundamentales del producto, como lo son el ver, crear y guardar las tarjetas de consejo, hacer preguntas, responder a las dudas y comentar, calificar o reportar los consejos de otros usuarios.

- Privilegios de acceso y seguridad:

Usuarios regulares: Tienen acceso limitado a ciertas funciones y datos, según su rol y responsabilidades dentro del producto, aquí se incluyen los usuarios básicos y los usuarios avanzados.

- Nivel de experiencia:

Sin medalla: Ninguna ventaja

Menos de 10 consejos y puntuación promedio sea igual o menor a 3.4 estrellas

Cobre: Los consejos y las respuestas de los usuarios de cobre tienen más prioridad de aparición que los de los usuarios sin medalla.

Más de 10 consejos publicados y puntuación promedio sea mayor a 3.5 estrellas y menor a 4

Plata: Los consejos y las respuestas de los usuarios de plata tienen más prioridad de aparición que los de los usuarios de cobre.

Más de 15 consejos publicados y puntuación promedio entre 4 y 4.5 estrellas

Oro: Los consejos y las respuestas de los usuarios de oro tienen más prioridad de aparición que los de los usuarios de plata.

Más de 20 consejos publicados y puntuación promedio mayor a 4.5 estrellas

IV. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

Creacion y publicacion de una carta:

Descripción: Un usuario registrado puede crear y publicar una carta de consejo, añadiendo el consejo y la pregunta que soluciona el consejo.

Acciones iniciadas y comportamiento esperado:

1. El usuario debe dirigirse a la sección de *create*.
2. En la sección *create* se dirige a *publicar consejo*
3. En *publicar consejo* el usuario debe escribir la pregunta a la que se le da solución con el consejo y el consejo,
4. Una vez hecho esto el usuario oprime en publicar consejo, ahí se muestra en pantalla un mensaje de confirmación.
5. Al confirmar se crea la nueva carta, haciendo que esta nueva carta de consejo entre en la cola de cartas de solución.

Requerimientos funcionales:

- Si el usuario oprime publicar consejo sin rellenar adecuadamente los espacios de consejo y pregunta del consejo no podrá publicar el consejo.
- Si el usuario ha sido suspendido temporalmente y se encuentra en periodo de suspensión no podrá realizar esta acción.
- En el campo de la pregunta que da solución al consejo deberá incluir los símbolos ‘¿’ y ‘?’

Eliminación de una carta (por el usuario que la creó)

Descripción: Un usuario registrado podrá eliminar una carta de consejo que él mismo haya creado, borrándose de la cola de cartas de consejo.

Acciones iniciadas y comportamiento esperado:

1. El usuario debe dirigirse a la carta creada por el, que desee eliminar.
2. Una vez en esta debe dar en la opción de Eliminar carta y ahí se muestra en pantalla un mensaje de confirmación.
3. Una vez confirmada la acción, la carta será borrada del sistema.

Requerimientos funcionales:

- Esta acción es definitiva y no se puede restablecer.
- La carta será eliminada de los Guardados de otros usuarios.

Consulta de Tarjetas

Descripción: Permite al usuario consultar todas las tarjetas almacenadas en el sistema, las tarjetas se muestran en el orden en que fueron ingresadas.

Acciones iniciadas y comportamiento esperado:

1. El usuario selecciona la opción de "Consultar Tarjetas" en el menú
2. El sistema recorre la lista encadenada y muestra en pantalla cada tarjeta con sus detalles (título, descripción, etiquetas, calificación).

Requerimientos funcionales:

- Las tarjetas deben mostrarse con todos sus atributos, de manera legible para el usuario.

- El sistema debe recorrer la lista encadenada de principio a fin, mostrando cada tarjeta.

Búsqueda de Tarjetas por Etiquetas

Descripción: Permite al usuario buscar tarjetas específicas según sus necesidades basadas en etiquetas asignadas a ellas

Acciones iniciadas y comportamiento esperado:

1. El usuario debe dirigirse a menú
2. El usuario debe seleccionar la opción de "agrupar Tarjeta por etiqueta" y proporciona la etiqueta deseada
3. El sistema agrupa las tarjetas en conjuntos disjuntos y permite al usuario explorar las tarjetas dentro de cada conjunto
4. El sistema busca en la lista encadenada todas las tarjetas que contengan la etiqueta especificada y las muestra al usuario

Requerimientos funcionales:

- El sistema debe permitir al usuario ingresar una etiqueta para buscar.
- El sistema debe recorrer la lista encadenada y verificar si la etiqueta está presente en cada tarjeta.
- Implementar la estructura de conjuntos disjuntos para organizar las tarjetas por etiquetas.
- Todas las tarjetas que coincidan con la etiqueta deben mostrarse en pantalla.

Ordenamiento de Tarjetas por Calificación

Descripción: Ordena las tarjetas según su calificación utilizando un montículo (heap), y permite al usuario ver las tarjetas mejor calificadas

Acciones iniciadas y comportamiento esperado:

1. El usuario debe dirigirse al menú
2. El usuario debe seleccionar la opción "Ver Tarjetas Mejor Calificadas".
3. El sistema extrae y ordena las tarjetas usando un montículo y muestra las tarjetas en orden descendente de calificación

Requerimientos funcionales:

- El sistema debe implementar un algoritmo de montículo (heap) para ordenar las tarjetas según su calificación.
- Mostrar las tarjetas en pantalla, comenzando por la mejor calificada.
- Permitir al usuario ver las tarjetas ordenadas y consultar detalles adicionales si es necesario.

Sugerencia de Tarjetas Similares Usando Grafos

Descripción: El sistema sugiere tarjetas similares utilizando un grafo que modela las relaciones entre tarjetas y etiquetas, las sugerencias se generan al encontrar nodos (tarjetas) cercanos en el grafo que compartan varias etiquetas o tengan alta relevancia

Acciones iniciadas y comportamiento esperado:

1. El usuario accede a una tarjeta desde la interfaz.
2. El sistema analiza el grafo, donde los nodos representan tarjetas o etiquetas y las aristas representan las relaciones entre ellas (por ejemplo, compartir etiquetas).
3. El sistema recorre el grafo buscando tarjetas similares basadas en la proximidad o fuerza de las conexiones

con la tarjeta actual, priorizando aquellas con mayor cantidad de etiquetas compartidas o relevancia.

4. El sistema muestra al usuario tarjetas sugeridas que están relacionadas a la tarjeta actual, basadas en la similitud de etiquetas según las conexiones del grafo

Requerimientos funcionales:

- El sistema debe utilizar un **grafo** donde los nodos representen **tarjetas y etiquetas**.
- Las **aristas** deben conectar tarjetas con etiquetas o entre ellas si comparten etiquetas.
- El sistema debe implementar un **algoritmo de búsqueda en grafos** (como BFS o DFS) para recorrer nodos y encontrar tarjetas similares.
- Las **tarjetas sugeridas** deben mostrarse basadas en la cantidad de etiquetas compartidas o la relevancia de las conexiones en el grafo.

Creación de Usuario

Descripción: Esta funcionalidad permite a los usuarios registrarse en el sistema proporcionando la información necesaria.

Acciones iniciadoras y comportamiento esperado:

1. El usuario accede a la página de registro.
2. Completa el formulario con su información personal (nombre de usuario, correo electrónico, contraseña, etc.).
3. Envía el formulario.
4. El sistema valida la información ingresada.
5. Si la validación es exitosa, se crea la cuenta del usuario y se le proporciona acceso al sistema.
6. Se muestra un mensaje de confirmación al usuario.

Requerimientos Funcionales:

- El formulario de registro debe incluir campos obligatorios para nombre de usuario, correo electrónico y contraseña.
- El sistema debe mostrar mensajes de error si estos campos no están completos o si el formato del correo electrónico es incorrecto.
- El sistema debe verificar que el nombre de usuario y la dirección de correo electrónico no estén ya registrados en el sistema.
- Si hay duplicados, el sistema debe mostrar un mensaje de error indicando que el nombre de usuario o correo electrónico ya están en uso.
- La contraseña debe cumplir con ciertos criterios de seguridad (longitud mínima, uso de caracteres especiales, etc.).
- El sistema debe mostrar un mensaje de error si la contraseña no cumple con los criterios establecidos.
- Después de un registro exitoso, el sistema debe enviar un correo electrónico de confirmación al usuario.

Eliminación de Usuario

Descripción: Esta funcionalidad permite a los usuarios eliminar permanentemente sus cuentas del sistema.

Acciones iniciadoras y comportamiento esperado:

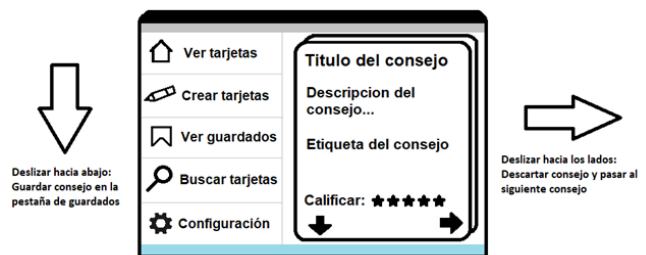
1. El usuario accede a la configuración de su cuenta.
2. Busca la opción de eliminar cuenta.
3. Confirma la eliminación de su cuenta.
4. El sistema elimina permanentemente la cuenta del usuario y todos sus datos asociados.
5. Se muestra un mensaje de confirmación al usuario.

Requerimientos Funcionales:

- El sistema debe proporcionar una opción clara y fácil de encontrar para eliminar la cuenta del usuario.
- Antes de eliminar la cuenta, el sistema debe solicitar una confirmación explícita del usuario.
- El sistema debe mostrar un mensaje de advertencia indicando que la eliminación de la cuenta será permanente y que todos los datos asociados se perderán.
- Después de eliminar la cuenta, el sistema debe cerrar la sesión automáticamente y redirigir al usuario a la página de inicio de sesión.

V. DESCRIPCIÓN DE LA INTERFAZ DE USUARIO PRELIMINAR

Ver tarjetas En la sección ver tarjetas aparecen las cartas de consejo (donde contiene el título del consejo, la descripción del consejo, la etiqueta y la calificación) y da la posibilidad a calificar el consejo, además al deslizar hacia abajo la tarjeta (u oprimir la tecla hacia abajo) se guarda la tarjeta, y al deslizar hacia el lado (u oprimir la tecla hacia los lados) se sigue a la siguiente tarjeta de consejo y se descarta la anterior, por lo que se implementa una estructura de Cola en la visualización de consejos.



Crear tarjeta: llevará a una pestaña donde se puede asignar un título al consejo, una descripción al consejo y etiqueta de este.

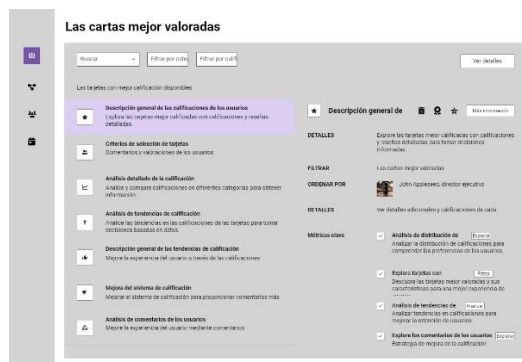


Ver tarjetas guardadas: Muestra las cartas de consejos que se han guardado, de forma similar a lo mostrado en *ver tarjetas*, pero mostrando primero las cartas de consejo guardadas más recientemente, ya que los en guardados se implementa una estructura de Pila/Stack

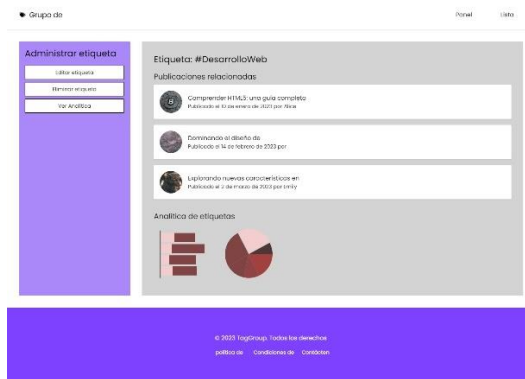
Buscar tarjeta consejo: Da la opción de buscar un consejo mediante una palabra clave y muestra al igual que en *ver tarjeta* una lista con los consejos que coincidan con lo buscado.

Actualizar o eliminar tarjeta consejo: Buscada la tarjeta y con los permisos necesarios (haber creado la carta de consejo) el usuario puede realizar cambios en la tarjeta de consejo o eliminar la tarjeta de consejo, con una interfaz similar a la de *crear tarjeta*.

Visualización de Tarjetas Mejor Calificadas: La pantalla muestra una lista ordenada de tarjetas según su calificación, con opciones para ver detalles adicionales y aplicar filtros, muestra las tarjetas en orden descendente según su calificación incluyendo también título, calificación, y una breve descripción de cada tarjeta.



Agrupación por Etiquetas: La pantalla de Agrupación por Etiquetas permite al usuario ver y gestionar tarjetas organizadas por sus etiquetas, facilita la visualización de tarjetas asociadas a etiquetas específicas y ofrece opciones para filtrar y manejar la información agrupada, muestra una lista de todas las etiquetas disponibles en el sistema incluye nombre de cada una y la cantidad de tarjetas asociadas.



Sugerencias de Tarjetas Similares Usando Grafos: La pantalla de *Sugerencias de Tarjetas Similares* permite al usuario ver y explorar tarjetas que están relacionadas con la tarjeta actual mediante un grafo que representa las conexiones entre tarjetas y etiquetas. Facilita la identificación de tarjetas asociadas a etiquetas similares y ofrece una manera intuitiva de descubrir contenido relevante. Además, presenta una lista de las tarjetas sugeridas, que incluye el nombre de cada tarjeta y la cantidad de etiquetas compartidas con la tarjeta de interés, ayudando al usuario a encontrar fácilmente contenido relacionado y relevante.

VI. ENTORNOS DE DESARROLLO Y DE OPERACIÓN

Se desarrollará una aplicación de escritorio para el sistema operativo Windows, utilizando el lenguaje de programación Java en su versión 8, con la actualización 401. Además, se empleará GitHub como plataforma para el control de versiones y la colaboración en el desarrollo del proyecto.

Se manejo el siguiente github: [NQM765/ProyectoEstructuras: Foro de consejos para la transición de vida adolescente a vida adulta \(github.com\)](https://github.com/NQM765/ProyectoEstructuras_Foro_de_consejos_para_la_transicion_de_vida_adolescente_a_vida_adulta)

VII. DISEÑO, IMPLEMENTACIÓN Y APLICACIÓN DE LAS ESTRUCTURAS DE DATOS

Lista: Lista es un clase que representa una cola implementada como una lista encadenada de nodos, estos nodos son de tipo genéricos, por lo que la lista puede contener elementos de cualquier tipo especificado. Para acceder y mantener los datos almacenados dentro de la lista encadenada se utiliza la interfaz Iterator sobre la misma, creando un objeto de la lista de tipo Iterator que itera sobre los elementos de la Lista.

Tarjeta: El aplicativo, con el objetivo de mostrar una interfaz reconocible para usuarios jóvenes (público objetivo del aplicativo) e inspirada en los videos cortos de formato vertical, implementa la clase Tarjeta, en esta aparece el título, descripción, etiqueta y calificación de un consejo. Por lo que sobre la clase Tarjeta gira gran parte del aplicativo, pues la Tarjeta como elemento de la clase Lista, es decir una Lista

encadenada de Tarjetas, es la forma como se le presenta al usuario los consejos (como si se tratara de un mazo de cartas), un usuario hace una nueva instancia de Tarjeta para crear un consejo, y el usuario guarda una Tarjeta si desea guardar el consejo.

Pila: La Pila es una clase que representa un Stack hecho con arreglos de tipo genérico. Una instancia de esta clase se utiliza para la implementación de guardados, en esta el usuario puede guardar cartas, y posteriormente acceder a ellas, siguiendo la metodología LIFO donde la última carta que guarda el usuario es la primera que ve al acceder a sus tarjetas guardadas.

Menu: La clase Menu contiene el ciclo de utilización de un usuario, es decir las opciones para la utilización del software, que consisten en la creación de tarjetas, la búsqueda de tarjetas, la actualización de tarjetas, guardar tarjetas, calificar tarjetas, la consulta de tarjetas y la consulta de tarjetas guardadas.

Montículos: Un montículo es una estructura de datos especializada que satisface la propiedad de heap, donde cada elemento cumple con una relación de prioridad con respecto a sus hijos en el prototipo, se utiliza un heap para gestionar las tarjetas mejor calificadas, permitiendo extraer de manera eficiente la tarjeta con la mayor calificación

Conjuntos Disjuntos: Los conjuntos disjuntos son una estructura de datos que maneja particiones de un conjunto de elementos, soportando operaciones de unión y búsqueda, en el prototipo, se utilizan conjuntos disjuntos para agrupar tarjetas según sus etiquetas, permitiendo una gestión eficiente de las tarjetas por categorías

VIII. PRUEBAS DEL PROTOTIPO Y ANÁLISIS COMPARATIVO

Para las pruebas del prototipo, se tomaron las funciones de búsqueda de tarjeta, eliminación de tarjeta y mostrar todas las tarjetas, y se hizo el seguimiento del tiempo mediante la clase `System.currentTimeMillis()`; obteniendo los siguientes resultados:

- 10 mil datos heap

```
Prueba maxHeap con 2 hijos:
Tiempo maxHeap con 2 hijos: 110 ms
```

```
-----
```

```
Prueba maxHeap con 3 hijos:
Tiempo maxHeap con 3 hijos: 79 ms
```

- 10 mil datos conjuntos

```
Tiempo sin compresión de caminos: 2900
Tiempo con compresión de caminos: 6320 ms
```

- 10 mil datos AVL

```
Prueba con Árbol de Búsqueda Binaria (BST):
Tiempo de inserción en BST: 882 ms
Tiempo de ordenación en BST: 0 ms
```

```
Prueba con Árbol AVL:
```

```
Tiempo de inserción en AVL: 1260 ms
Tiempo de ordenación en AVL: 2 ms
```

Para realizar un análisis asintótico de la tabla dada, primero necesitamos identificar la tendencia de crecimiento del tiempo de ejecución en función del tamaño de entrada (en este caso, el número de tarjetas). Luego, podemos determinar la complejidad asintótica de cada funcionalidad en términos de la notación Big O.

Dado que las mediciones están dadas para diferentes tamaños de entrada (10 mil, 100 mil, 1 millón, etc.), es útil calcular la relación entre el tamaño de entrada y el tiempo de ejecución. Para simplificar, tomaremos la relación entre el tiempo de ejecución y el tamaño de entrada más grande (100 millones en este caso).

Dado que la complejidad asintótica se enfoca en el comportamiento a medida que el tamaño de entrada se acerca al infinito, utilizamos la relación entre el tiempo de ejecución y el tamaño de entrada más grande para determinar la complejidad asintótica.

1. Búsqueda de tarjeta:

- Parece que el tiempo de ejecución crece linealmente con respecto al tamaño de entrada. Entonces, la complejidad asintótica sería $O(n)$, donde n es el número de tarjetas.

2. Eliminación de tarjeta:

- De manera similar, el tiempo de ejecución también parece crecer linealmente con respecto al tamaño de entrada. La complejidad asintótica también sería $O(n)$.

3. Mostrar todas las tarjetas:

- Al igual que las dos funcionalidades anteriores, el tiempo de ejecución crece linealmente con respecto al tamaño de entrada. Por lo tanto, la complejidad asintótica sería $O(n)$.

4. Ver Tarjetas Mejor Calificadas

- La complejidad esperada es $O(\log n)$ para inserción y extracción el tiempo de extracción de la tarjeta mejor calificada crecerá logarítmicamente con el tamaño de los datos.

5. Ver Sugerencias de Tarjetas Similares

- La complejidad esperada para la sugerencia de tarjetas es $O(\log n)$, lo que significa que el tiempo necesario para encontrar y mostrar tarjetas similares crecerá logarítmicamente con el tamaño de los datos. Esto asegura que, a medida que aumenta la cantidad de tarjetas en el sistema, la búsqueda de sugerencias se mantenga eficiente y rápida, proporcionando una experiencia de usuario fluida.

En resumen, todas las funcionalidades parecen tener una complejidad asintótica lineal $O(n)$, donde n es el número de tarjetas.

IX. INFORMACIÓN DE ACCESO AL VIDEO DEMOSTRATIVO DEL PROTOTIPO DE SOFTWARE

Enlace web (URL) para tener acceso al video demostrativo del prototipo de software funcional:

<https://youtu.be/RR8efT0ut9E>

X. ROLES Y ACTIVIDADES

INTEGRANTE	ROL(ES)	ACTIVIDADES REALIZADAS (Listado)
Laura Sofia Vargas Rodriguez	Investigadora	Trabajo en prototipo
		Requerimientos funcionales
	Observadora	Trabajo en diapositivas
		Trabajo en video
Nicolas Quezada Mora	Líder	Trabajo en prototipo
		Requerimientos funcionales
	Coordinador	Coordinar reuniones
		Trabajo en video
Jerónimo Bermúdez Hernández	Técnico	Trabajo en prototipo de interfaz
		Trabajo en prototipo
	Coordinador	Requerimientos funcionales
		Trabajo en diapositivas
Jorge Isaac Alandete Diaz	Diseñador	Trabajo en prototipo de interfaz
		Requerimientos funcionales
	Experto	Trabajo en prototipo
		Trabajo en video

XI. DIFICULTADES Y LECCIONES APRENDIDAS

Se esperaba que el proyecto se completara en un plazo más breve; sin embargo, el desarrollo del prototipo y su posterior entrega enfrentaron complicaciones y detalles imprevistos que resultaron en una extensión del plazo original. Estas dificultades nos llevaron a reevaluar nuestros tiempos y expectativas.

Otro desafío significativo fue el dominio del lenguaje Java, especialmente para los miembros del equipo que lo abordaban por primera vez. Esta curva de aprendizaje afectó las entregas, ya que el entendimiento del lenguaje y sus particularidades requirió tiempo adicional, impactando en el cronograma general del proyecto.

Durante este proceso, también enfrentamos la dificultad de crear una interfaz intuitiva y atractiva, además, la implementación de nuevas estructuras de datos, como grafos y tablas hash, presentó sus propios retos, sin embargo, estas experiencias nos permitieron adquirir conocimientos valiosos

sobre la utilización de estructuras de datos secuenciales y no secuenciales en proyectos, destacando su importancia y cómo facilitan el manejo de la información.

XII. REFERENCIAS BIBLIOGRÁFICAS

- [1] D. Hartmann y T. T. Swartz, «The New Adulthood? The Transition to Adulthood from the Perspective of Transitioning Young Adults», *Advances In Life Course Research*, vol. 11, pp. 253-286, ene. 2006, doi:10.1016/s1040-2608(06)11010-2.
- [2] D. Marzana, A. M. Pérez -Acosta, E. Marta, y M. Gonzalez, «La transición a la edad adulta en Colombia: una lectura relacional», *DOAJ (DOAJ: Directory Of Open Access Journals)*, oct. 2010, [En línea]. Disponible en: <https://doaj.org/article/a632b57fb9da4390ac2ca21dc8e7fe9d>
- [3] M. M. D. Santos, B. L. Queiroz, y A. P. De AndradeVerona, «Transition to adulthood in Latin America:1960s-2010s», *Revista Brasileira de Estudos dePopulação*, vol. 38, pp. 1-11, ago. 2021, doi:10.20947/s0102-3098a0161.
- [4] O. De Oliveira y M. M. Salas, «Desigualdades sociales y transición a la adultez en el México contemporáneo», *Papeles de Poblacion*, vol. 14, n.o 57, pp. 117 -152, sep. 2008, [En línea]. Disponible en: <http://www.scielo.org.mx/pdf/pp/v14n57/v14n57a6.pdf>
- [5] J. J. Arnett, «Concepciones de la transición a la edad adulta: perspectivas desde la adolescencia hasta la mediana edad.», *Journal Of Adult Development*, vol. 8, n.o 2, pp. 133-143, ene. 2001, doi: 10.1023/a:1026450103225.