

## Crea tu pensum (CTP)

**Yeira Liseth Rodríguez, Andrés Felipe Castro, Juan Manuel Pérez, Andrés David Ramírez,  
Maria Camila Sanchez**

**No. de equipo de trabajo: {8}**

### I. INTRODUCCIÓN

Generalmente los estudiantes que mejor gestionan su tiempo son los que mejores resultados obtienen, de hecho “entre el alumnado de bajo rendimiento, la planificación de tareas resulta poco efectiva. Las razones se encontrarían en un cálculo erróneo sobre el tiempo que les llevará la realización de ciertas actividades.” (Garzón et al, 2018). Lo anterior evidencia que podrían existir herramientas que ayuden a un mejor manejo de las tareas, con base a horarios específicos. Sobre este panorama se desarrolla Crea Tu Pensum (CTP), una herramienta que también recopila las distintas experiencias tediosas que pueden sufrir los estudiantes al ahora de saber que asignaturas inscribir cada que inicia semestres teniendo en cuenta todos los prerrequisitos, dificultad, número de créditos entre otros criterios, por esto es que CTP basada en las carreras universitarias, específicamente las pertenecientes a la Universidad Nacional de Colombia de la sede Bogotá, que permite la creación de múltiples planes de estudio según la preferencia del usuario, con base a un numero de materias y créditos personalizables que estructuran un diseño en el que los estudiantes pueden basar su vida universitaria, CTP parte de la idea que un crédito es el equivalente a 48 horas de estudio, para que la carga académica este acorde a las actividades que realice la persona. Por otro lado, facilita que alguien que repite una materia, debe nivelar o tiene otras actividades aparte de la universidad como trabajar, realizar cursos, estudiar etc., tenga siempre opciones con el fin de llevar a cabo y organizar su vida a futuro.

Finalmente, luego de dar un poco de contexto de Crea Tu Pensum observamos que permite y da pie para el uso de las estructuras de datos, ya que, a pesar de que CTP será aplicable al pregrado de ingeniería de sistemas (en un principio), se podrá ampliar a más planes de estudio, más sedes de la Universidad Nacional e incluso extenderse a universidades que se encuentren a lo largo de Colombia, lo cual aumentaría la cantidad de datos almacenados además de mostrar la efectividad, buen uso y los diversos beneficios de utilizar implementaciones originales con las distintas estructuras de datos existentes.

### II. DESCRIPCIÓN DEL PROBLEMA A RESOLVER

El problema que busca resolver CTP es facilitar la vida de los estudiantes de la Universidad Nacional de Colombia que quieran organizar su vida universitaria, utilizando como base sus planes de estudio para que lo personalicen con base a la carga académica que ellos prefieran por semestres, logrando

que la comunidad universitaria cuente con un mejor manejo de su tiempo.

### III. USUARIOS DEL PRODUCTO DE SOFTWARE

**Usuario final:** Persona interesada crear su propio plan de estudios mediante los recursos que se ofrecen en nuestro programa. tendrá permisos mínimos, los cuales solo le permiten acceder a el producto final y las aplicaciones relacionadas con este como lo son el SIA u otras fuentes de información relacionadas al proyecto.

**Entidad contratadora:** Entidad que contrata el software para sus estudiantes, la cual, está encargada de entregarlo a cada usuario final, y solucionar todos los problemas de logística administrativa. Tendrá permiso de distribución del código únicamente a los estudiantes que lo requieran.

**Experto informático:** Persona que gestionará y apoyará a los usuarios finales para que estos accedan al programa. Tendrá permisos de alteración a la interfaz del programa, pero no podrá alterar su contenido.

**Experto Aliado:** Persona encargada de suministrar nueva información al programa si es necesario, como la adición de una nueva carrera, materias etc. Sus permisos serán cercanos a absolutos, pero dependerá de la validación de la entidad contratadora.

### IV. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

CTP describe un número limitado de funcionalidades (por el momento) para que el usuario se familiarice con la forma en que funciona el software. Las primeras funcionalidades presentan una vista general del programa, tres específicamente, donde el usuario primero podrá documentarse de las materias que hay en su carrera y saber a nivel general con que datos funciona el programa, otra funcionalidad a la que se le hace especial énfasis es el generador de pensum pues es la base de la idea de CTP, por ultimo hay una funcionalidad que muestra lo ya hecho en la segunda y permite que el usuario detalle las mallas que más le interesen dentro de las generadas.

- *Consultar asignatura*
- *Descripción:*
  - Consultar asignatura funciona de manera similar al Sistema de Información Académica de la universidad,

mostrando una descripción básica de la materia que quiera buscar el usuario, mostrando número de créditos y las condiciones de prerequisites.

- *Acciones iniciadoras y comportamiento esperado:*

1. El estudiante o persona perteneciente a la Universidad Nacional de Colombia sede Bogotá selecciona en el menú principal la funcionalidad de buscar asignatura, que es representada con el número 1 (uno) en la consola.
2. Posteriormente, el usuario ingresa el código de la asignatura a buscar de su preferencia, de esta forma el programa busca dentro de todas las asignaturas previamente guardadas según el semestre.
3. Se muestran tres recuadros de información correspondiente a la materia, el nombre, número de créditos y la condición de prerequisite que le corresponde. Aquí el software busca dentro de la lista que contiene las materias del pregrado y muestra la información que tiene sobre esta que fue almacenada previamente.

- *Creador de mallas*

- *Descripción:*

- Creador de mallas toma en cuenta los parámetros ingresados tanto por el usuario (número de créditos por semestre) como los predeterminados del programa (número de semestres máximo que se definió en 20), para que, de esta manera, tomando todos los prerequisites de las materias en una carrera, se generen múltiples posibilidades en las que el usuario se puede acoger para organizar su carga académica dentro de la vida universitaria-

- *Acciones iniciadoras y comportamiento esperado:*

1. El estudiante o persona perteneciente a la Universidad Nacional de Colombia sede Bogotá selecciona en el menú principal la funcionalidad de generar pensum, que es representada con el número 2 (dos) en la consola.
2. Posteriormente, el usuario ingresa los parámetros requeridos por el programa para generar las mallas, estos son el número de créditos por semestre, a partir de este momento el programa analiza dentro de unas pilas que se crearon de prerequisites con las materias correspondientes, y empieza a generar numerosos planes de estudio donde se organizan las materias de forma distinta, adentrándonos más en el funcionamiento se hizo uso de dos

arreglos, uno de índices que representan los semestres, donde el número que se encuentra en cada índice representa el número de materias que se agregan a partir de otro arreglo que contiene las materias de la carrera en cuestión, de esta manera resultan materias estructuradas en distinto semestre, distinto orden dentro de la malla etc.

3. Se almacenan cada una de las mallas dentro de un arreglo dinámico que crece a medida que se agrega cada uno de los pensum creados.

- *Mostrar malla*

- *Descripción:*

- Mostrar malla permite que el usuario vea en pantalla una de las mallas que anteriormente genero la funcionalidad creador de mallas, de esta forma puede observar de manera detallada la que más le intereso con base a los parámetros que ingreso previamente.

- *Acciones iniciadoras y comportamiento esperado:*

1. El estudiante o persona perteneciente a la Universidad Nacional de Colombia sede Bogotá selecciona en el menú principal la funcionalidad de mostrados de pensum, que es representada con el número 3 (tres) en la consola.
2. Posteriormente, el usuario ve en pantalla todas las mallas en forma de matriz que almaceno anteriormente un arreglo el generador de pensum.
3. Dentro de la consola se una de las mallas que se crearon con base a los parámetros que ingreso el usuario y los predeterminados para que funcione CTP, para versiones posteriores se busca una mejor y más amplia interacción del usuario para que pueda editar, eliminar o cambiar las mallas a su gusto.

#### *Requerimientos funcionales:*

Las funcionalidades mínimas sobre los datos que se manejen deben prever operaciones de:

- Creación
- Actualización
- Eliminación
- Consulta total de los datos
- Búsqueda parcial de datos
- Ordenamiento
- Almacenamiento

Teniendo en cuenta esta lista de requerimientos se pensó en una estructura de datos recurrente y fácil de manejar dada la naturaleza del proyecto, el arreglo dinámico, de esta manera se hicieron tres implementaciones distintas de la misma estructura para luego realizar las pruebas correspondientes que

se mostraran en una sección más adelante, aparte del arreglo dinámico se utilizaron pilas así como listas enlazadas para que representen los prerrequisitos de cada materia y en general, la estructura de todo el plan de estudios.

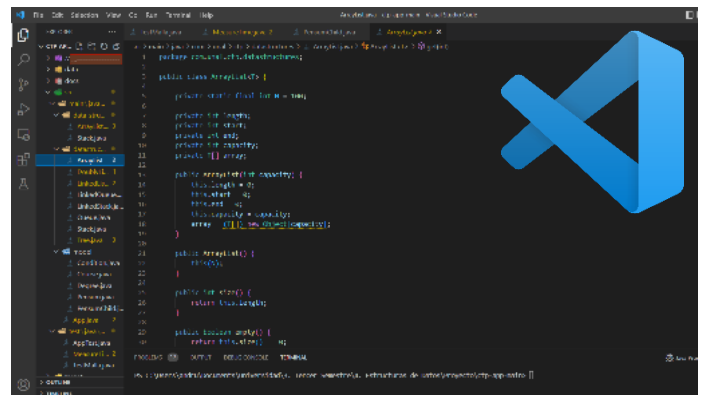
### Descripción de la interfaz de usuario preliminar

Crea Tu Pensum busca integrar todas sus funcionalidades en una misma pantalla, para facilitar el uso del software, aquí se pueden realizar distintas acciones que van desde elegir la sede, facultad y carrera o modificar el número de créditos y número materias (donde están las letras a y b) hasta seleccionar que materias ya se han visto simplemente mostrando la malla predeterminada que enseña la universidad a sus estudiantes.

Todo esto busca mostrar todas las capacidades de CTP en un solo lugar, de manera que solo al final cuando se creen las numerosas mallas, sea el mismo usuario capaz de elegir, modificar y finalmente guardar el plan de estudios que más se acomode a sus gustos.

### V. ENTORNOS DE DESARROLLO Y DE OPERACIÓN

El entorno de desarrollo que se utilizó para el proyecto fue Visual Studio Code, junto con todas las extensiones necesarias para realizar las pruebas correspondientes a la entrega, a su vez, se clono el proyecto en el IDE de eclipse para tener una vista general de las clases y objetos trabajados. Por último, para poner en funcionamiento el software se utilizará de manera provisional la terminal del entorno de desarrollo, para posteriormente transportarlo a la web y que a CTP se pueda acceder desde una página.



Por otro lado, el hardware sobre el que operara será principalmente ordenadores con sistema operativo Windows, así, una vez se resuelva la compatibilidad dentro de los equipos de trabajo de los integrantes del proyecto se puedan pensar en diversos sistemas operativos que se generalicen y sean accesibles desde una única página web que representara todo CTP.

### VI. PROTOTIPO DE SOFTWARE INICIAL

```

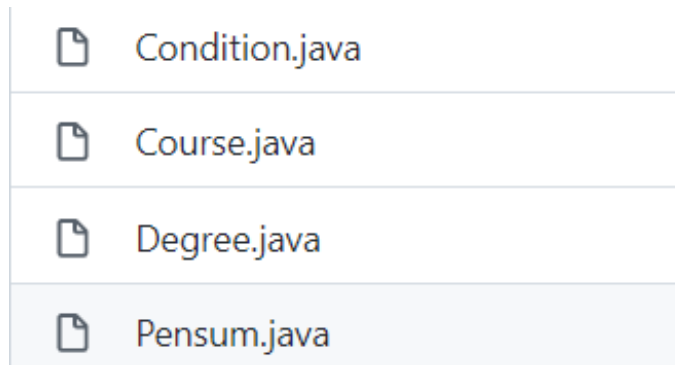
root
├── README.md
├── src
│   ├── my.packages // Carpeta donde se almacena el código fuente
│   ├── my.packages // Paquete principal del proyecto (kernel)
│   ├── Main.java // Clase principal que ejecuta la aplicación. Ejemplo. Java: clase Main.java
│   └── ... // Otros archivos java con el código fuente de la app
├── docs
│   ├── Entrega (Plantilla).pdf // Documento plantilla con la información de la entrega requerida
│   └── ... // Otros documentos según se requiera
├── data // Carpeta con los datos de prueba del proyecto
├── dist // Carpeta en donde se almacenan los archivos que ejecutan la aplicación
│   ├── proyectoEntrega1.jar
│   ├── proyectoEntrega2.jar
│   └── proyectoEntrega3.jar
├── lib // Carpeta con las librerías requeridas en la aplicación
└── Otros archivos... // Otros archivos de ejecución y configuración necesarios.
    // Ejemplo: project.xml, build.xml (ant), run.sh, run.bat (ejecución), etc.
  
```

Repositorio correspondiente en GitHub:

<https://github.com/dre-us/ctp-app>

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

Para el diseño y la estructuración de nuestro proyecto hicimos uso de la programación orientada a objetos, ya que esta nos permite dividir el proyecto en diferentes capas las cuales contienen de manera funcional cada una de las clases que forman el proyecto. Lo más importante de esta estructuración es la manera como estas se relacionan, de modo que las clases a continuación se relacionan e interactúan de la siguiente manera:



1. La clase Condition, al ser la clase que permite crear el objeto que representa un prerrequisito necesario para ver un curso específico se relacionan por multiplicidad con la clase Course, debido a que esta representa los cursos obligatorios de una carrera, por lo que un arreglo de objetos de la clase Condition será un atributo de un objeto de la clase Course.

2. Similar al modo en el que se relacionan las clases Condition y Course, la clase Course y Degree mantienen una relación de multiplicidad ya que un objeto de la clase Degree representa una carrera específica de las ofertadas por la Universidad.

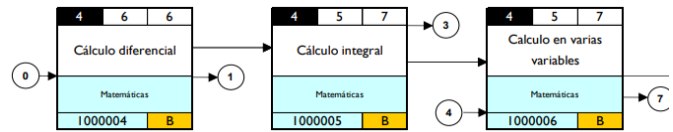
Por lo tanto, entre otros atributos de esta clase, uno de ellos será una lista de objetos de la clase Course, el cual representará todos aquellos cursos obligatorios de aquella carrera.

3. Un objeto de la clase Pensum representa una posible distribución por semestres de la lista de cursos (respetando siempre sus prerrequisitos).

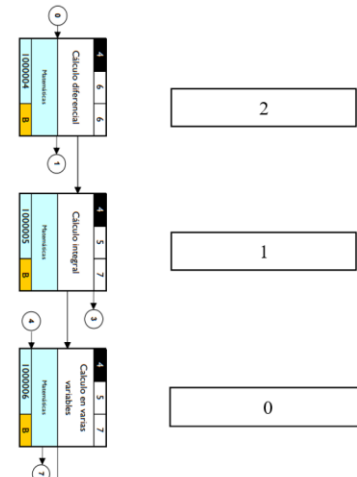
De esta manera, al igual que las anteriores clases, la clase Degree y la clase Pensum se relacionan por multiplicidad, pero esta de una manera particular a las anteriores ya que en esta un atributo de un objeto Pensum no es una lista de objetos de la clase Degree, sino que en esta comparten cada uno de los objetos de la lista Course, porque todos los cursos que aparecen en Degree deben aparecer en Pensum, respetando así el cumplimiento de todos los cursos para completar determinada carrera.

Una representación de lo que pasa en estos tres puntos donde se relacionan las clases del proyecto se muestra a continuación con algunos ejemplos y muestras graficas con el plan de

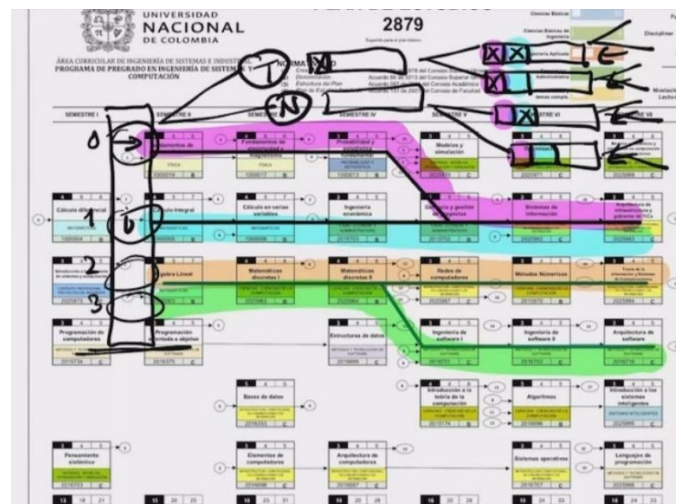
estudios que normalmente se ve de la carrera de ingeniería de sistemas.



Entonces, una de las condiciones de calculo en varias variables es calculo integral y a su vez una de las condiciones de cálculo integral es calculo diferencial, las condiciones aumentan si una materia requiere que se vea más de una materia para llegar a ella, ahora, esto se realiza para que con una pila vista de la siguiente manera:



Una vez generamos las pilas con los conjuntos de materias vamos a la funcionalidad de generador de pensum, donde ahora tenemos múltiples pilas que representan los prerrequisitos de las materias:



Con esta estructura somos capaces de ver un pensum como un conjunto de pilas que tiene prerrequisitos, en este caso la imagen muestra las distintas pilas con distinto color y como estas se van agregando a un pensum aleatorio generado por *generador de pensum*.

Ahora, utilizamos nuestras implementaciones de arreglo dinámico, “arraylist”, las cuales importamos a cada funcionalidad a la hora de realizar las pruebas correspondientes, sin embargo, y cabe resaltar, que se implementaron todas las estructuras de datos vistas hasta el momento para desarrollar CTP con cualquiera de ellas, para observar si en un futuro una estructura de datos distinta tiene mejor rendimiento dentro del funcionamiento del software.

### VIII. PRUEBAS DEL PROTOTIPO Y ANÁLISIS COMPARATIVO

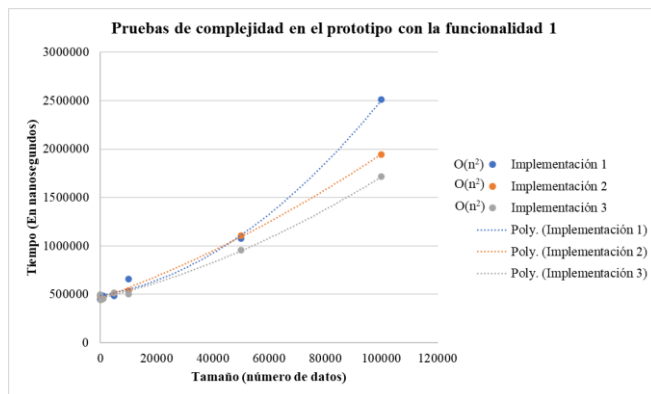
A continuación, presentamos las pruebas que se hicieron con las implementaciones de arraylist, la estructura de datos en común que tenían todas las funcionalidades con los siguientes números de datos.

#### Para la funcionalidad 1 Consultar asignatura:

Se compararon tres implementaciones distintas de arraylist, mostrando una complejidad de  $n^2$  llegamos a la conclusión que se pueden utilizar métodos más efectivos cuando se tenga la posibilidad de utilizar estructuras de datos multidimensionales y a su vez de buscare una mayor efectividad en entregas posteriores.

Tamaño (número de datos)	Tiempo (En nanosegundos)		
	Implementación 1	Implementación 2	Implementación 3
10	451500	483800	497200
100	485800	478600	444300
500	485600	457100	451700
1000	485500	457800	456700
5000	487100	513600	516900
10000	656800	538600	503400
50000	1078800	1104600	956400
100000	2512300	1946200	1713500

#### • Análisis asintótico:



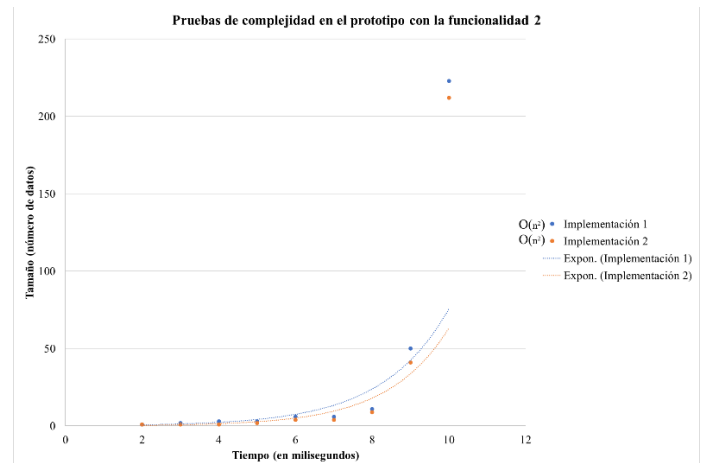
#### Para la funcionalidad 2 Creador de mallas:

En el creador de mallas se compararon dos de las implementaciones de arraylists y su respectivo uso en las implementaciones de pilas correspondientes. Incluso, en crear malla se utilizó recursión lo que explica el alto costo computacional cuando se intentan realizar las distintas combinaciones de mallas con base a los cursos que se le ingresan, en este caso lo máximo que soporto el método con 10 cursos, generando una gran cantidad de mallas.

Se busca que en la entrega siguiente se realicen infinitas combinaciones de mallas que se acoten con base a más parámetros que se añadirán en futuras funcionalidades de CTP.

Tamaño (número de datos)	Tiempo (en milisegundos)	
	Implementación 1	Implementación 2
2	1	1
3	2	1
4	3	1
5	3	2
6	6	4
7	6	4
8	11	9
9	50	41
10	223	212

#### • Análisis asintótico:

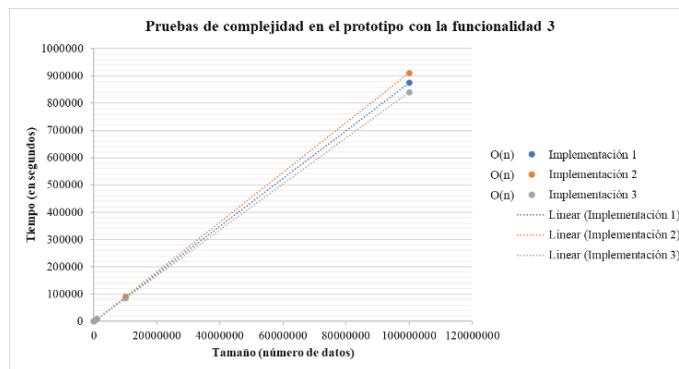


### Para la funcionalidad 3 *Mostrar malla*:

Mostrar malla muestra un funcionamiento bastante general en forma lineal, a pesar de que no es ideal es el mejor rendimiento que se tuvo dentro de las tres funcionalidades. Aquí fue posible ingresar gran cantidad de datos, llegando a los 100 millones, lo cual muestra un buen panorama de esta funcionalidad que espera ser incluso más interactivo y a la vez rápido que contribuya a un mejor rendimiento de CTP.

Tamaño (número de datos)	Tiempo (En segundos)		
	Implementación 1	Implementación 2	Implementación 3
10000	87.951	91.873	84.721
100000	874.313	913.313	842.565
1000000	8747.756	9131.822	8418.845
10000000	87475.031	91101.102	84192.465
100000000	874752.973	911256.213	840521.758

- Análisis asintótico:**



A modo de conclusión, las pruebas coinciden en algunos aspectos la una de la otra, gracias a la forma en que se implementó cada funcionalidad o el número de datos que se ingresaban teníamos resultados distintos. En detalle podemos observar cómo tanto en la funcionalidad 1 y 3 la implementación del arraylist más rápida fue la tercera, claro que en consultar asignatura este cambio fue más evidente mientras que en mostrar malla mostraba un comportamiento muy parecido.

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

[https://youtu.be/I\\_eDFJD\\_3c](https://youtu.be/I_eDFJD_3c)

### X. ROLES Y ACTIVIDADES

Integrante	Rol(es)	Actividades fundamentales
Andrés Felipe Juan Manuel	Lideres	Organizaron el equipo y pensaron lógicamente el funcionamiento del proyecto.
Maria Camila	Coordinador	Mantuvo contacto con el grupo a través de correos y mensajes.
Andrés Felipe	Experto	- Coordinó el traslado del proyecto a GitHub. - Organizó el CTP de manera que el equipo trabajara en conjunto.
Andrés David Juan Manuel	Investigador	- Dieron alternativas a la resolución inicial del proyecto. Investigaron manera y métodos distintos de realizar las funcionalidades.
Yeira Liseth	Observador(a)	Se conectó de manera periódica para ver el avance de cada uno de los integrantes del proyecto
Andrés David	Animador	Motivó al grupo cuando había dificultades al final de la entrega
Maria Camila	Secretario(a)	- Documentó lo acordado en reuniones presenciales y virtuales. - Realizó una lista de tareas de lo que se debía realizar para la entrega preliminar de CTP
Yeira Liseth Andrés Felipe Juan Manuel Maria Camila Andrés David	Técnico(s)	- Se dividieron y trabajaron en el desarrollo de software para finalmente unirlos en un producto final que integraba cada una de las partes que conformaban a CTP.

### XI. DIFICULTADES Y LECCIONES APRENDIDAS

Una de las dificultades que tuvimos en esta entrega fue ponernos de acuerdo con la forma en que creábamos los algoritmos en cada funcionalidad, como cada uno imaginaba algo distinto y esto podía chocar para cuando era momento de integrar cada una de las partes del proyecto, por esta razón aprendimos a tener una mejor comunicación una vez estábamos más cerca a la fecha de entrega, organizando nuestras ideas al mismo tiempo que creamos un formulario de Google siempre abierto a sugerencias con el fin de llegar a un

consenso de las mejores y mas eficientes maneras de abordar cada uno de los problemas que debemos resolver dentro del proyecto los cuales irán incrementando en dificultad a medida que CTP amplia sus carreras, facultades, sedes e incluso universidades disponibles en un futuro.

Al momento de trabajar en equipo son muchos los factores que pueden dificultar la cooperación entre los integrantes, incluso considerando el hecho de actualmente se puede hacer de manera virtual y presencial. Con el desarrollo de esta primera parte del proyecto llegamos a la conclusión de que la manera óptima de trabajar es estructurando el proyecto en diferentes partes y repartiendo tareas a cada uno de los integrantes (o subgrupos) para así superar posibles dificultades en el acuerdo de fechas de encuentro.

## XII. REFERENCIAS BIBLIOGRÁFICAS

- [1] Umerenkova, Angélica Garzón y Flores, Javier GilGestión del tiempo en alumnado universitario con diferentes niveles de rendimiento académico. Educação e Pesquisa [online]. 2018, v. 44 [Accedido 27 marzo 2022], e157900. Disponible en: <<https://doi.org/10.1590/S1678-4634201708157900>>. Epub 04 Set 2017. ISSN 1678-4634. <https://doi.org/10.1590/S1678-4634201708157900>.