

Diseño y Programación Orientada a Objetos – Sección 6

Jerónimo Vargas Rendon - 202113305

Nicolas Camargo Prieto – 202020782

Documento Reflexivo de los Proyecto

Introducción

A lo largo de la clase de Diseño y Programación Orientada a Objetos, se realizó un estudio sumamente teórico, pero también con una aplicación práctica de diferentes teorías de construcción de proyectos desde el lenguaje de programación Java. Desde un inicio, se estableció que se harían 3 proyectos en base a 3 marcos teóricos importantes: el diseño y planeación de un proyecto de código usando UML como lenguaje estandarizado, la implementación de una interfaz gráfica, y el manejo organizacional del proyecto con base a WBS, manejo de errores y excepciones.

En consecuencia, a continuación, se realizará un análisis en detalle de las diferentes entregas que se realizaron para la construcción de los tres proyectos. Dicho análisis se obtendrá mediante la explicación lógica de los procesos que se realizaron para llegar a la construcción final y entrega de los proyectos, con detalle acerca de los principales ajustes, problemáticas y aspectos positivos que se encontraron a la hora de desarrollar los requisitos principales.

Aspectos Positivos

Desde un punto de vista general, se identificó que, pese a ciertas dificultades afrontadas a lo largo de los proyectos, se llegó a un mejor entendimiento y manejo de Java, teniendo en cuenta sobre todo el uso de librerías que ayudan a la reutilización y a los procesos de creación de software. Además, gracias a esto también se pudo identificar la gran importancia de la planeación en el proceso de diseño de un proyecto de código, el manejo de posibles errores y la anticipación de los diferentes problemas que pueden surgir cuando se está desarrollando un proyecto.

En adición, se evidencio a lo largo del desarrollo que se logró en gran medida mantener un proyecto organizado y que cumplió con los procesos de planeación que se establecieron desde un inicio para cada entrega.

Aspectos problemáticos de los procesos

Por otro lado, a lo largo de todo el proyecto se identificaron varias falencias y problemas importantes, los cuales tuvieron consecuencias importantes en el desarrollo. Desde un primer momento, la adaptación a Java, así como, a la IDE de Eclipse sumado a Github fueron difíciles de manejar, pues se trata de un entorno muy distinto a lo que se venía trabajando y represento un reto adquirir la mayoría del conocimiento necesario fuera de la teoria vista en clase para el desarrollo de los proyectos. Además, los principales errores en la planeación del código resultaron en problemas importantes que tuvieron que ser corregidos

oportunamente, pues en base a lo que se plantea desde la primera entrega se establecen las bases de las siguientes, por lo cual, una planeación débil, ineficiente o carente de información resultaría en un proyecto incompleto y desordenado.

Procesos de cada entrega por proyecto

Ahora bien, se explicarán a detalle las principales dificultades, oportunidades y retos que se encontraron, así como, los procesos que se realizaron para cada entrega.

Proyecto 1

En este primer momento en el cual se pidió evidenciar como se pone en práctica los conocimientos estudiados en clase hay varios aspectos a mencionar con las diferentes entregas que se realizaron.

- **Entrega 1 Análisis**

En esta entrega se puede decir que el proceso que realizamos fue bastante acertado ya que se lograron identificar todas las responsabilidades del dominio, para luego así tener una comprensión exhaustiva que permite plantear un fuerte diseño de la lógica para así iniciar a implementar una solución. Lo anterior se logró conseguir gracias a una discusión en equipo que permitió tener una buena comprensión de la tarea propuesta y así poder discernir entre sus componentes aplicando los conocimientos brindados por el curso.

- **Entrega 2 Diseño**

Una vez analizada la propuesta se pidió crear un diseño de la lógica del dominio para que asfixiara todas las responsabilidades propuestas y aspectos identificados en el análisis. En esta entrega podemos decir que surgieron algunas dificultades en el sentido de relacionar el conocimiento del curso con el uso de herramientas para desarrollar diagramas UML. Estas dificultades se dieron ya que era bastante confuso relacionar detalles conceptuales con representación gráfica dentro del diagrama, como por ejemplo saber cómo representar si un atributo es privado o público, o yéndonos a algo más complejo, el cómo representar la relación de los objetos de la forma que uno quiere comunicar, es decir saber si hay una relación de agregación, composición o herencia de una interfaz o clase abstracta y demás.

- **Entrega 3 Implementación**

El comienzo de este proceso fue bastante complicado ya que uno llega con desconocimiento del IDE eclipse, del lenguaje y la lógica de una programación orientada a objetos.

Adaptarse a la programación orientada objetos mediante Java fue un proceso que dificultó el dar el primer paso para desarrollar la lógica del dominio, pero gracias al tiempo dedicado adicional a las clases para estudiar la documentación del lenguaje y del IDE se logró tener un conocimiento para poder implementar la lógica del dominio diseñada en esta primera etapa que resulto ser funcional. Y con relación a la programación orientada a objetos el curso brinda los conocimientos para estructurar un objeto con atributos y métodos que sean requeridos.

Proyecto 2

Este segundo momento del proyecto dio vuelta a todo lo que se había venido trabajado en la carrera, ya que se introduce conocimiento con relación a buenas prácticas de diseño e implementación de interfaces gracias desarrolladas por una librería de java.

- Entrega 1 Diseño

Para esta primera entrega era simplemente plantear un bosquejo el cual fue funcional para las responsabilidades que indicaba el dominio.

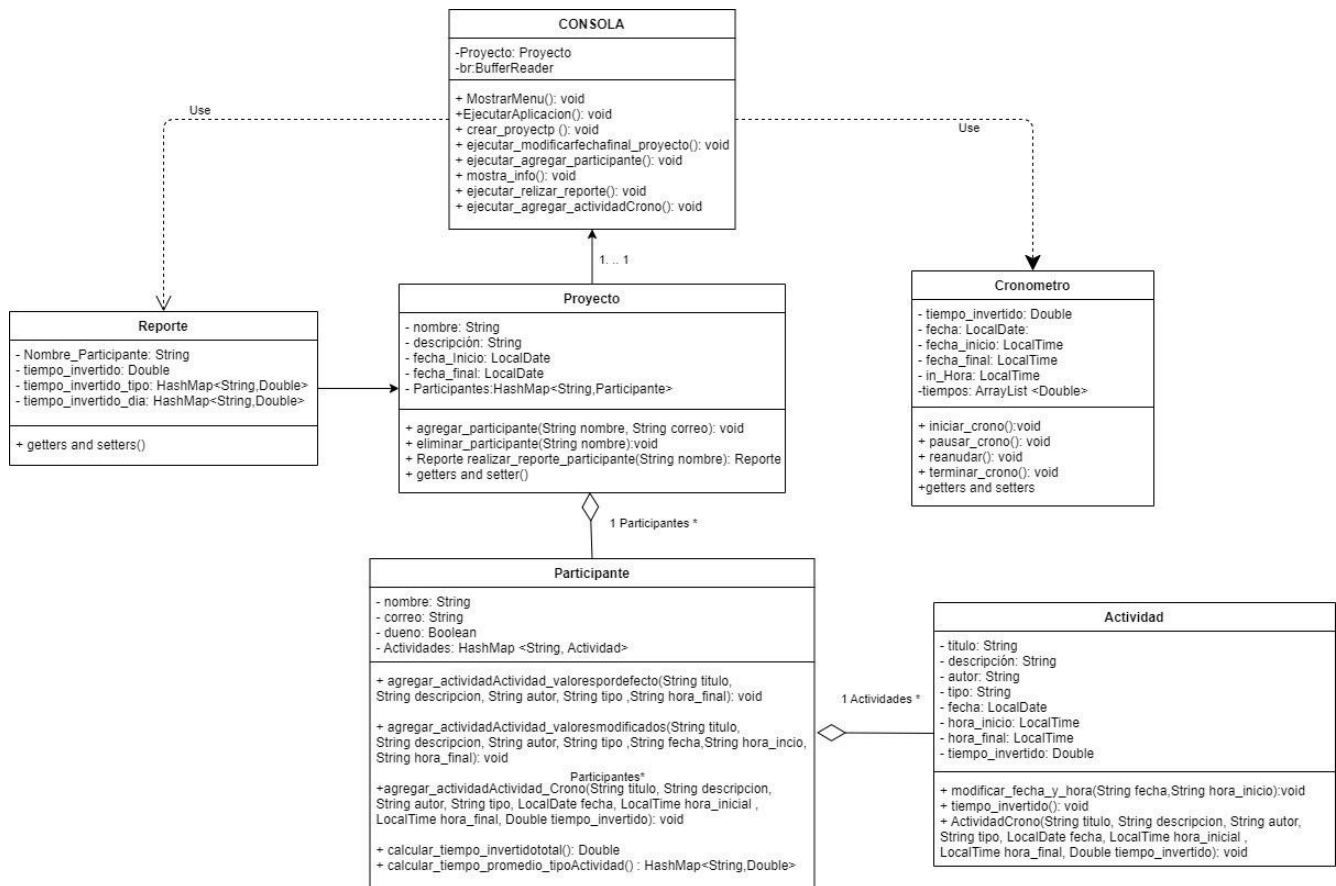
- Entrega 2 Implementación

En esta entrega de implementación claramente se nos hizo más complejo ya que nunca se había trabajado en desarrollos de interfaces de usuario hechas con código. El hecho de que la interfaz sea programable hizo que la tarea fuera ardua y nos obligara a estudiar documentación de la librería de Swing para poder interactuar con el usuario de la mejor manera y además como se había requerido.

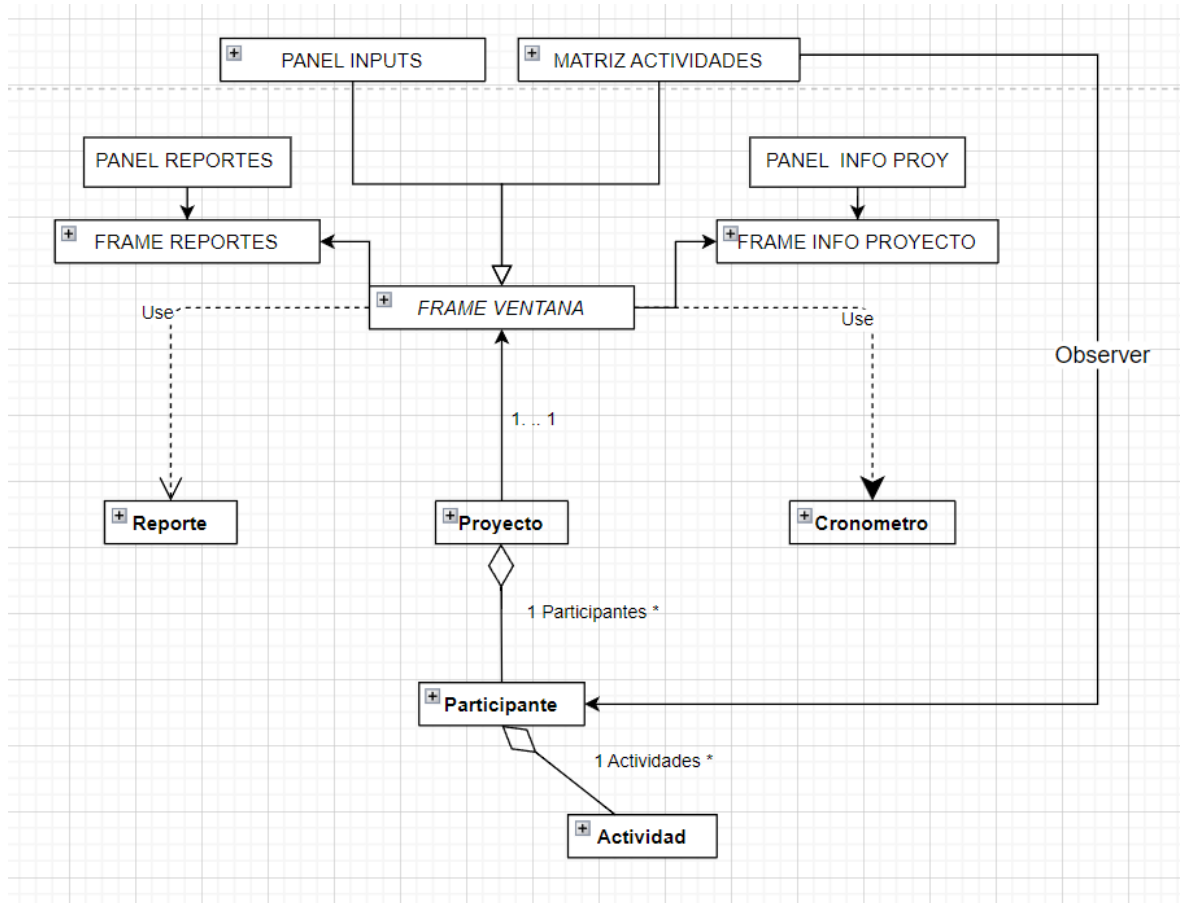
En el momento de hacer la implementación se podría afirmar que lo más sencillo fue estructurar el panel de inputs el cual recibía toda la información de del Usuario. Pero luego la responsabilidad de generar una matriz dinámica y los Reportes por participantes se complicó mucho.

En el caso de los reportes se tomó la decisión de generar un nuevo frame cada vez que se generara un reporte para que así este se pudiera con la información que hay en el momento del proyecto y dar sus estadísticas, la decisión anterior funciona bastante eficiente e interactivo para el usuario.

Por otro lado, la tarea más difícil para realizar la matriz de actividades, ya que al ser dinámico la única alternativa que pensamos fue usar el patrón Observer el cual permitió que un Panel encargado de la matriz observara el comportamiento de la lógica del Dominio. Es Aquí en esta parte del proceso dónde se descubrió que la relación de dependencia de los objetos dificultó buscar una alternativa para aplicar este patrón y relacionar el panel de matriz con las actividades agregadas a la base de datos.



Como pueden ver en el diagrama de la lógica de Propuesto en el Primer proyecto, la única forma de acceder a las actividades del proyecto es a través de los participantes y cada participante contiene varias actividades y este objeto el de participante el que tiene el método para notificar las actividades, es por esto que se aplicó el patrón observer entre la matriz y los participantes, y para poder solucionar el problema de que la cantidad de participantes no es fija, toco desde la interfaz agregarle a todos los participantes existentes el componente de observable para notificarle al observer. Como se muestra en la relación de a continuación.



De la misma forma, el proceso que se realizó para este segundo proyecto tuvo ciertas irregularidades, pues lo que fue planeado en una primera instancia en el proceso de diseño tuvo que ser efectivamente corregido en el proceso de implementación. Esto debido a que, el uso de ciertas herramientas tecnológicas en la implementación de la interfaz gráfica con Swing tuvo ciertos problemas en la ejecución y administración de los componentes de la interfaz. De tal manera, la adición de ciertos aspectos planeados fue más difícil de lo que se anticipó y requirió de mayor tiempo.

Proyecto 3

Para este último monumento ya se tienen más desarrolladas las habilidades con relación al uso de las herramientas para implementar, pero al ser la última instancia es cuando se empiezan a notar las consecuencias de decisiones aplicadas anteriormente en el proceso del desarrollo del dominio.

- Entrega 1 Diseño

En esta entrega el diseño como en todo el semestre es más flexible y discutible, ya que es una propuesta de cómo puede realizarse la implementación y no necesariamente debe llegarse estrictamente a lo propuesto en el diseño, ya que durante la implementación pueden surgir

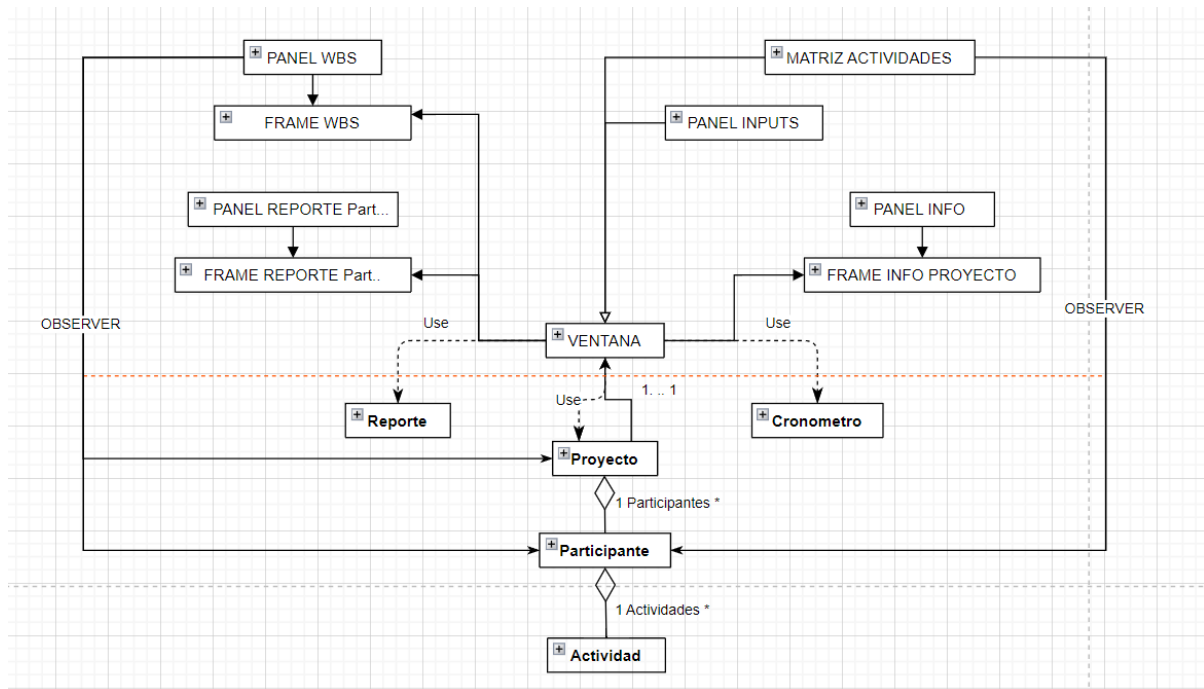
más iteraciones que permite cambiar el diseño inicial ya que posiblemente algunas decisiones no eran pertinentes.

- Entrega 2 Implementación

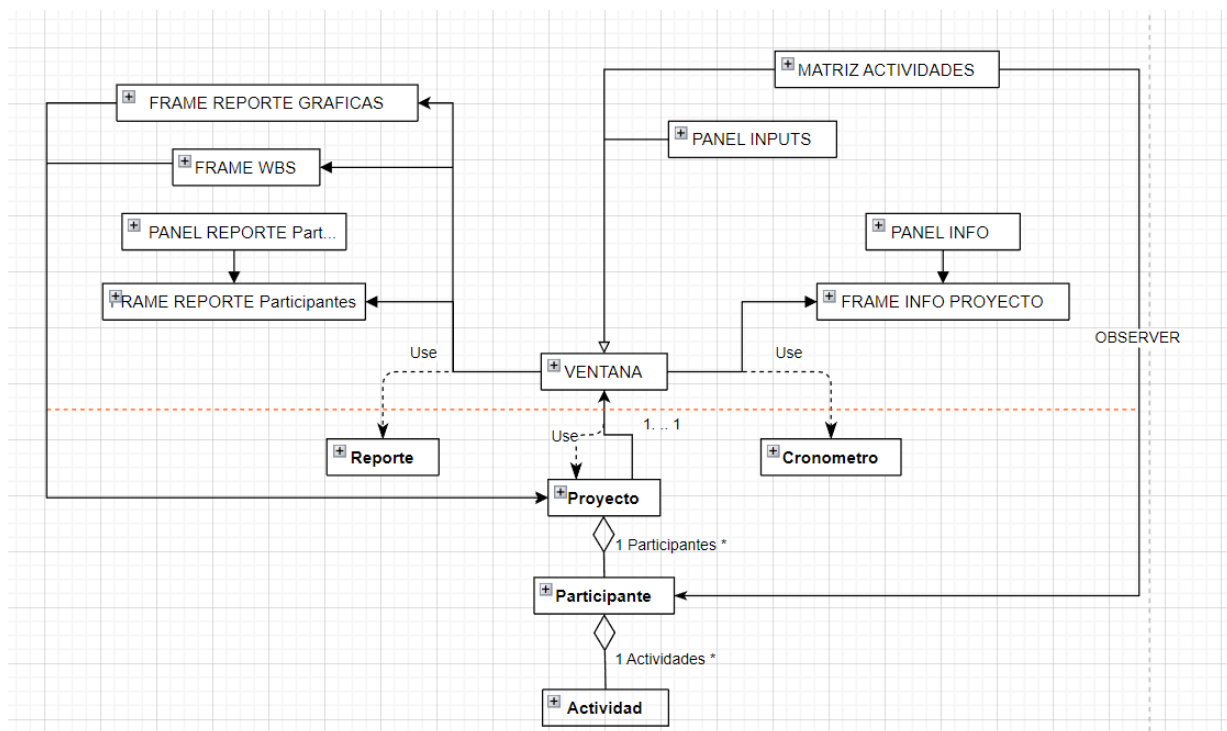
En el momento de implementar las ultima responsabilidades del dominio vemos que surgen bastantes inconvenientes por lo que se generan de nuevo más iteraciones ya que se detecta que las decisiones tomadas en el diseño no fueron del todo pertinentes. ya que se pide mostrar información del proyecto que este no generaban y ya la estructura de datos creada en la lógica limitaba los mecanismos para filtrar información. Es por este que por ejemplo para el caso del WBS no se logró detectar una alternativa para generar un wbs que fuera dinámico y se actualizar constantemente de forma visual (El patrón de observer se hallaba muy difícil de implementar), por ende, se optó por crear un Action Listener que permitiera visualizar el wbs en otro frame con el fin de que cada vez que este se construyera lo hiciera con los datos actuales de la base de datos de la lógica de negocio. Por otro lado, para generar los reportes gráficos se utilizó la librería de JFreeChart y se detectó que la alternativa de generarlos en vez de que fueran dinámicos era menos compleja y generaba menos acoplamiento, ya que simplemente se generan métodos en el proyecto que llevan la información a graficar.

A continuación, pueden comparar el modelo del Diseño en el Proyecto 3 y como fue el definitivo.

Diseño (UML) Entrega 1 Proyecto 3



Diseño Final (UML) Entrega 2 Proyecto 3



En los modelos se evidencia como se decidió no implementar el patrón observer entre el Frame WBS con las clases Proyecto y Participante, ya que no era fácil construir el WBS con el mecanismo de notificar acciones de estas clases. Por lo anterior se decidió usar el componente visual de JTree que se encuentra dentro de la Clase Frame WBS, ya que se comporta visualmente muy similar a lo que requería el dominio de visualizar un Work Breakdown Structure. De igual forma vemos el nuevo Frame de Reportes de Graficas que usa funcionalidades de Jfreechart para representar un análisis estadístico de los datos recopilados sobre el proyecto del cual se está monitoreando.

En esta etapa también se identificó que algunas de las responsabilidades pedidas en el Dominio eran un poco confusas e incluso hasta ambiguas, ya que inicialmente en la primera esta se pedía que el proyecto agregara actividades donde cada una tenía un tipo de tarea la cual desempeñaba, y ahora en esta entrega se dice que una tarea tiene un tipo, lo cual es confuso para filtrar información y mostrar la información que realmente pertenece a ese objeto. En conclusión, la estructura en la que el proyecto almacena la información hizo que la complejidad para buscar y calcular los datos pedidos por las responsabilidades aumentara.

Conclusiones y consideraciones finales

Ya para finalizar se puede decir que durante todo este proceso se desarrollaron muchas habilidades ligadas a los conceptos y teoría de la programación orientada a objetos, también se evidencio la importancia de los principios de diseño e implementación de patrones de diseño, ya que son principios y métodos estructurados por una comunidad de programadores que busca hacer que la sociedad se más eficiente y productivos en sus prácticas de desarrollo. Por otro lado, podemos decir que el curso es muy productivo obliga al estudiante a ser autodidacta para conocer herramientas que le permitan ir cumpliendo con las implementaciones requeridas para que al final tenga un conocimiento de herramientas para futuras ocasiones.

En caso de volver a repetir este proceso, yo le recomendaría a los que continúan con este curso que saquen tiempo para estudiar la herramienta de eclipse y su lenguaje Java, ya que se comporta y maneja principios muy diferentes en comparación a lo que se ha trabajado en Python dentro de los primeros cursos. Y una vez empiece el proceso trabaje bajo su criterio con relación a los conocimientos aprendidos en clase y no se preocupe en estrellarse, que eso hace parte de la creación, lo peor que pueda pasar es replantearse una solución y seguir creando que al final vera resultados y trabajara con el fin de ir en pro de los objetivos del curso.

Por último, es muy importante que se tengan en cuenta las recomendaciones dadas en clase acerca de las diferentes etapas de planeación, desarrollo y manejo de pruebas o errores, para que la construcción de su proyecto sea eficiente, dando uso de las diferentes herramientas suministradas y las técnicas de organización que se estudian. De manera que, su proyecto sea ordenado, limpio y con todas las consideraciones técnicas que se requieren.