



PROYECTO PRIMERA ENTREGA INGENIERIA DE SOFTWARE

TRICOTECH

ITI-CETP

Prof. [Andrés Aranda]





ÍNDICE

1. Relevamiento.....	3
1.1 Formulario relevamiento.....	3
2. Estudio de factibilidades.....	3
2.1 Factibilidad Operativa.....	4
2.2 Factibilidad Técnica	4
2.3 Factibilidad Económica	5
2.4 Factibilidad Legal	5
3. Paradigma de conformación del equipo.....	6
4. Especificación de Requerimientos	6
5. Implementación del modelo de desarrollo.....	6
6. Lógica del sistema.....	7
7. Documentación de inicio y planificación de proyecto.....	8
7.1 Planificación inicial.....	8
8. Paradigma de conformación en coordinación con UTULab.....	8
9. Diagrama de Gantt.....	9
10. Diagrama de PERT.....	10
11. Prototipado de la aplicación.....	11
12. Carta de presentación del equipo.....	12
13. Reglamento Interno del equipo.....	12
14. Creación repositorio Git.....	12



1. Relevamiento

Para el revelamiento es muy importante saber que le puedo preguntar al cliente y que necesidades este tiene, mediante la entrevista, la observación, en este caso la propuesta del proyecto ya está estructurada y entregada por la UTU con sus respectivos requerimientos.

Si lo que necesitamos no estuviera en la letra del proyecto le preguntaría al cliente:

1.1 Formulario relevamiento

- ¿Hay un reglamento base ya definido del juego o lo tenemos que desarrollar desde cero?
- ¿Deseas un modo de juego de seguimiento?
- ¿Cuáles son los objetivos del juego? ¿Cómo se gana?
- ¿Qué mecánicas tiene que tener el juego, por Ejemplo: draft, colocación de fichas, puntuación por zonas...
- ¿Cuántos jugadores pueden jugar a la vez el juego?
- ¿Cuál será la duración promedio de una partida?
- ¿El juego es una creación original o una adaptación del Draftosaurus ya existente?

Dentro de esas preguntas podemos ver un tipo de técnica aprendida en clase Ej.: Las Preguntas Abiertas que son las que permiten al cliente expresar libremente sus ideas y las Preguntas Cerradas que limitamos al cliente a contestar más concretamente sobre nuestra pregunta.

2. Estudio de factibilidades

Para hacer un estudio de factibilidades lo primero que debemos determinar es la factibilidad operativa, técnica, económica y legal del proyecto.

Preguntarnos ¿Es factible desarrollar este proyecto de Draftosaurus operativamente para nosotros? ¿Es Útil? ¿Contamos con los conocimientos necesarios?



2.1 Factibilidad Operativa

Contamos con los recursos necesarios básicos por lo menos, que es tener una computadora relativamente buena en casa e internet. Somos un grupo formado por 4 personas lo que hace más fácil la división de tareas si la organización es la correcta. Nuestro producto será amigable con el usuario, el juego es fácil de aprender y para toda la familia. Contamos con el tiempo necesario si no pasa nada extra que prohíba o impida la realización del mismo en tiempo y forma.

2.2 Factibilidad Técnica

En este caso es un proyecto web por lo que no necesitaremos crear una app y teniendo en cuenta eso tampoco necesitaremos internet para jugarlo local. El juego es tanto como para computadora, celular y Tablet. Para que el juego sea jugable en dichos dispositivos lo vamos a hacer **responsive** lo que lo hace adaptable a cualquier tipo y tamaño de pantallas

Software mínimo para realizar el juego en la siguiente tabla:

Tipo de software	Recomendado	Función principal
Sistemas Operativos	Windows 7 en adelante, macOS, Linux, Android, iOS	Entorno de desarrollo
Editor de Código	Visual Studio Code o similar	Escribir y editar HTML, CSS, JS
Servidor Local	XAMPP 8.X o servidor escolar	Ejecutar código PHP localmente (simula un servidor web)
Navegador Web	Chrome 60+, Mozilla Firefox 55+, Safari 10+, Brave y Edge actualizados 2017 mínimo. Prácticamente todos menos Explorer ya que está obsoleto y no soporta características modernas	Probar como se ve y funciona el juego
Control de Versiones	Git + GitHub	Guardar avances del proyecto y colaborar en equipo
Diseño gráfico	GIMP, Canva	Crear o editar fichas, tableros, fondos, etc.



Software mínimo para poder jugar:

Software mínimo para poder jugar es exactamente lo mismo que necesitamos nosotros los desarrolladores, pero omitiendo lo siguiente: Visual Studio Code, GitHub o parte del diseño gráfico como GIMP o Canva.

Hardware mínimo para poder desarrollar / jugar el juego en la siguiente tabla:

Componente	Mínimo requerido (funciona)	Recomendado
Procesador (CPU)	Intel Pentium 4 / AMD equivalente (1.5 GHz)	Intel i3 o superior (2 GHz o más)
Memoria RAM	2 GB	4 GB o más
Almacenamiento	500 MB libres (para sistema + navegador)	2 GB libres (si se instala servidor local)
Pantalla	Resolución 1024x768	1366x768 o superior
Conexión a internet	No es obligatoria si es local	Para documentación, librerías o jugarlo online
Placa de video	Grafica integrada estándar	Cualquiera por encima de lo estándar está muy bien porque el juego no está en 3D con ultra gráficos

2.3 Factibilidad Económica

Este proyecto se va a llevar a cabo sin fines de lucro y tampoco tendríamos la obligación de gastar dinero en impresiones, prototipos o materiales ya que la institución nos brinda lo necesario.

2.4 Factibilidad Legal

Al ser un juego ya creado con su reglamento y marca registrada nosotros no vamos a utilizar este juego con fines de lucro como ya mencionamos en la



anterior factibilidad ni para sustituir a la obra original, se desarrollará una versión propia, con diseño original, adaptada al contexto educativo

3. Paradigma de conformación del equipo

Este equipo está organizado por el conjunto entre el Paradigma de Programador jefe y el Paradigma Abierto.

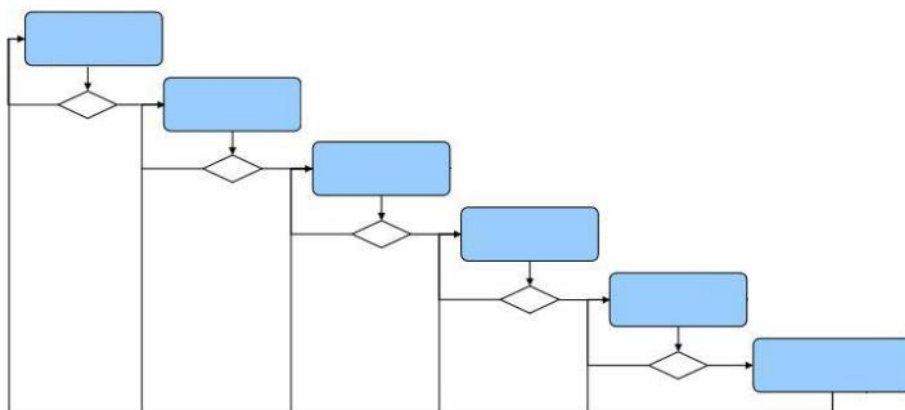
Elegimos Programador jefe porque en el equipo hay un líder que es la autoridad y es el Coordinador del equipo, también tenemos al ayudante del líder que en nuestro caso es el Subcoordinador del equipo y el resto del grupo que son dos integrantes que son quienes llevan a cabo las tareas delegadas por el Coordinador y/o Subcoordinador. Y el modelo de Paradigma Abierto porque en el equipo hay tareas que se llevan a cabo grupalmente en base a mucha comunicación y la toma de decisiones es consensuada.

4. Especificación de Requerimientos

https://drive.google.com/file/d/1i8bVbjCAsjNzkt1yiAOS8lULc1pjA5a/view?usp=drive_li nk

5. Implementación del modelo de desarrollo

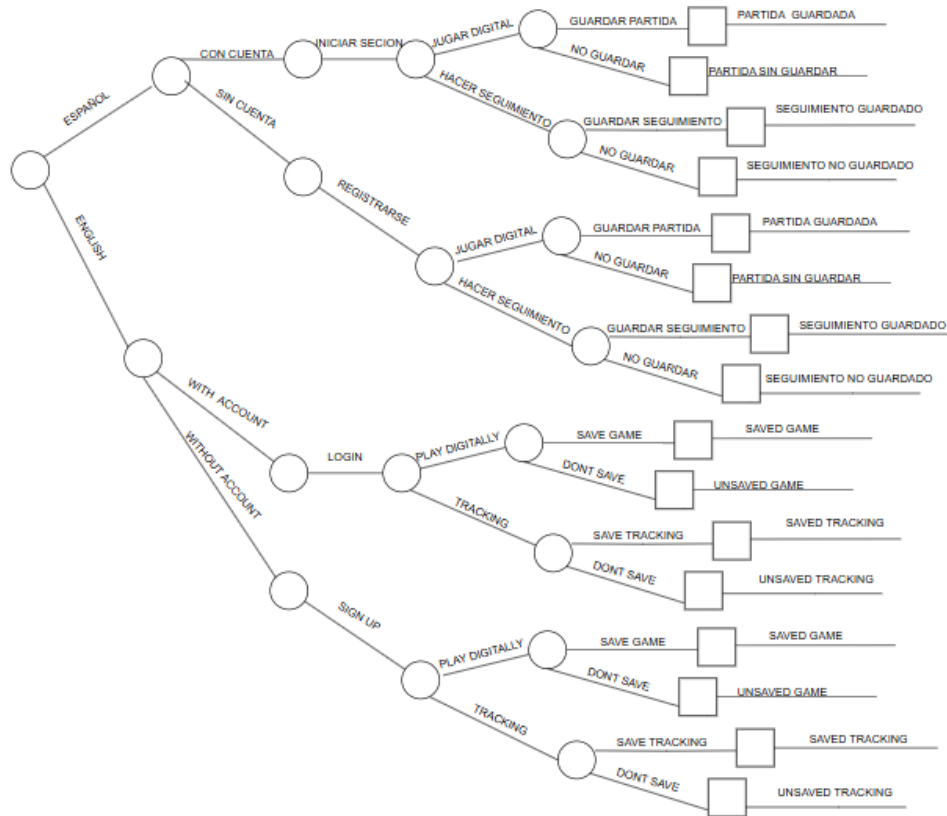
Para la realización de este proyecto se implementó una **metodología tradicional** basada en el **modelo en cascada** porque tiene una estructura sencilla que permite desarrollar el proyecto de forma secuencial, ordenando el trabajo en fases claramente diferenciadas como análisis, diseño, desarrollo, pruebas y mantenimiento. Esta estructura facilita la planificación, documentación, control del proceso y verificación de cada etapa antes de avanzar a la siguiente, este modelo es ideal para proyectos con requisitos bien definidos y estables como es en nuestro caso con el juego Draftosaurus.





6. Lógica del sistema

Árbol de decisión:



El siguiente árbol está generado a partir de las siguientes decisiones:

- Selección de idioma
- Si ya tiene cuenta o tiene que registrarse
- Va a jugar digital o va a hacer seguimiento?
- Guardar partida o no

Más adelante agregaremos más decisiones, pero en un principio sería como está detallado en la imagen.



7. Documentación de inicio y planificación del proyecto

7.1 Planificación inicial

Nuestra planificación inicial fue leer bien la letra cada uno de los integrantes del grupo para entender bien lo que había que hacer, para luego hacer una división de materias por cada uno de nosotros, el plan era cada uno de los integrantes tiene que hacer los requerimientos de 2 materias de la rama tecnológica con las que nos sintiéramos más cómodos y las materias de la rama común las haríamos en conjunto vía Discord o en persona. En el grupo de Discord y WhatsApp se sube material y las tareas realizadas. Nos planteamos objetivos a corto y largo plazo para las diferentes entregas. En las llamadas de Discord hacemos los requerimientos de las materias de la rama común.

Cuando ya tenemos los requerimientos completados de nuestras materias asignadas hacemos llamada en Discord y cada uno de los integrantes muestra su trabajo, explica como hizo en cada uno de los requerimientos y responde dudas de los compañeros. Una vez ya terminada la explicación se recopila toda la documentación para volver a leerla y que este la completa aceptación de todos los integrantes y así lograr las entregas en tiempo y forma.

7. Paradigma de conformación del equipo (Coordinación con UTULab)

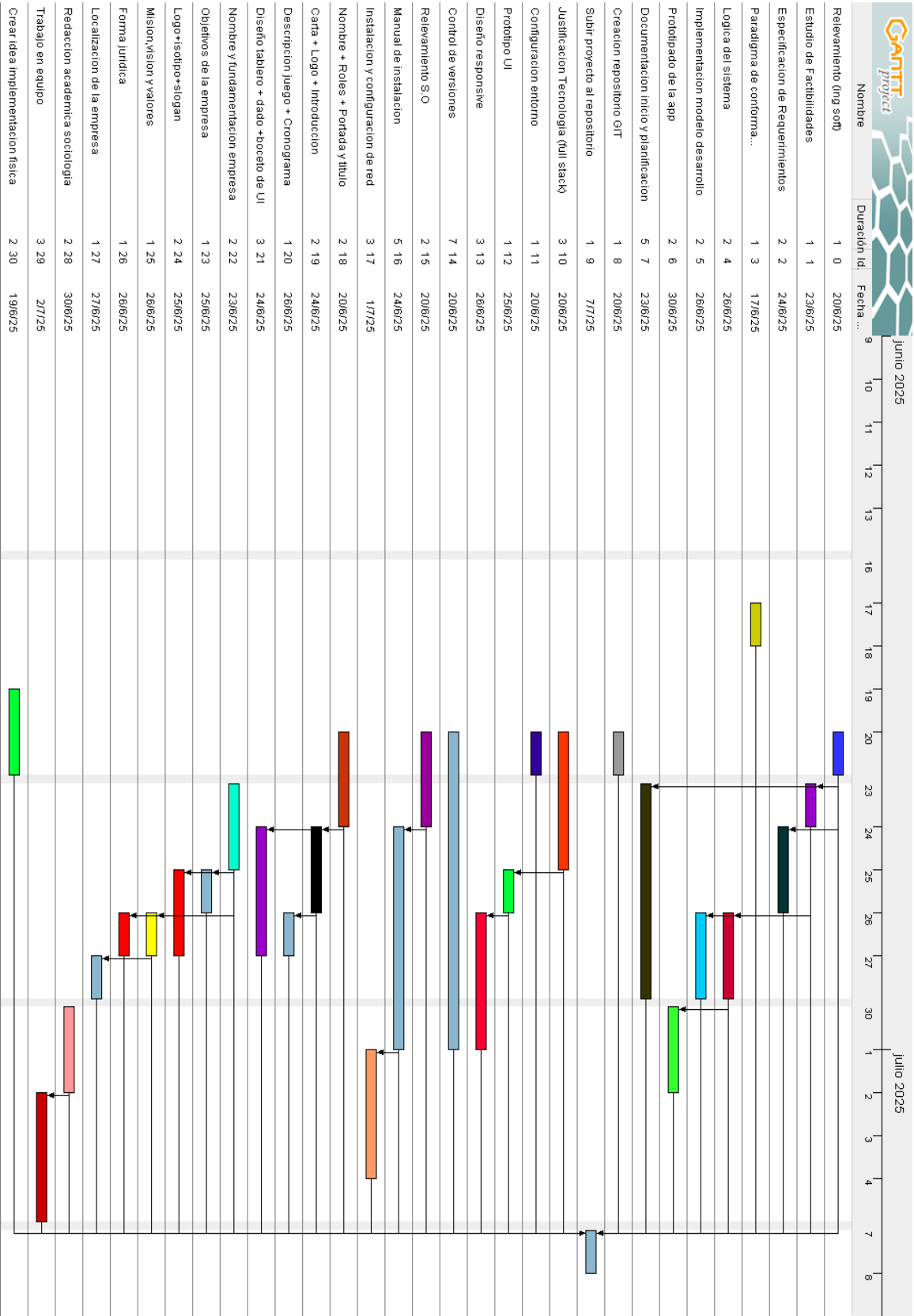
Como no hablamos aun de ningún paradigma en UTULab subo el mismo que para Ingeniería de Software.

Este equipo está organizado por el conjunto entre el Paradigma de Programador jefe y el Paradigma Abierto.

Elegimos Programador jefe porque en el equipo hay un líder que es la autoridad y es el Coordinador del equipo, también tenemos al ayudante del líder que en nuestro caso es el Subcoordinador del equipo y el resto del grupo que son dos integrantes que son quienes llevan a cabo las tareas delegadas por el Coordinador y/o Subcoordinador. Y el modelo de Paradigma Abierto porque en el equipo hay tareas que se llevan a cabo grupalmente en base a mucha comunicación y la toma de decisiones es consensuada.

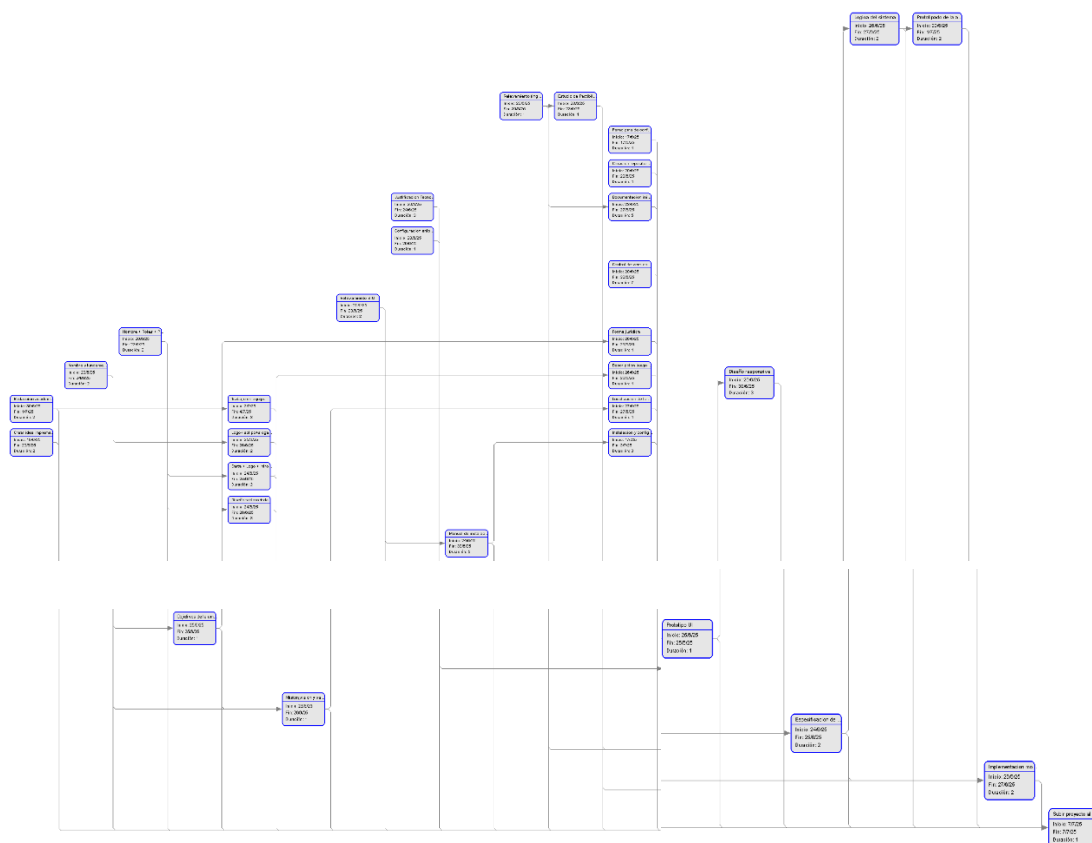


9. Diagrama de Gantt





10. Diagrama de PERT con GanttProject

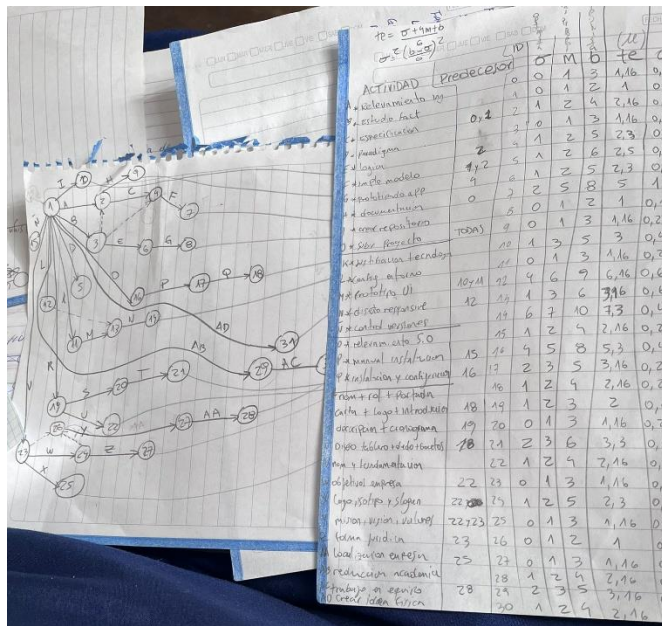


$$te = o + 4m + b / 6 \quad \sigma^2 = (b - o) / 6^2$$

	A	B	C	D	E	F	G	H
1	Actividad	Predecesor	ID	o	m	b	Te	σ^2
2	Relevamiento	x	0	0	1	3	1.16	0.25
3	Estudio fact	x	1	0	1	2	1	0.1
4	Especificación	0 y 1	2	1	2	4	2.16	0.25
5	Paradigma	x	3	0	1	3	1.16	0.25
6	Logica	2	4	1	2	5	2.3	0.4
7	Modelo imp	1 y 2	5	1	2	6	2.5	0.6
8	Prototipado	4	6	1	2	5	2.3	0.4
9	Documentación	0	7	2	5	8	5	1
10	Base de datos	x	8	0	1	2	1	0.1
11	Configuración	TODAS	9	0	1	3	1.6	0.25
12	Prototipo UI	x	10	1	3	5	3	0.4
13	Definición de requisitos	x	11	0	1	3	1.16	0.25
14	Control de versiones	10 y 11	12	4	6	9	6.16	0.6
15	Levantamiento de datos	12	13	1	3	6	3.16	0.6
16	Instalación	x	14	6	7	10	7.3	0.4
17	Control de portabilidad	x	15	1	2	4	2.16	0.25
18	Artefactos de desarrollo	15	16	4	5	8	5.3	0.4
19	Descripción de requisitos	16	17	2	3	5	3.16	0.25
20	Definición de requisitos	x	18	1	2	4	2.16	0.25
21	Definición de requisitos	18	19	1	2	3	2	0.1
22	Definición de requisitos	19	20	0	1	3	1.16	0.25
23	Definición de requisitos	18	21	2	3	6	3.3	0.4
24	Definición de requisitos	x	22	1	2	4	2.16	0.25
25	Definición de requisitos	22	23	0	1	3	1.16	0.25
26	Definición de requisitos	22	24	1	2	5	2.3	0.4
27	Definición de requisitos	22 y 23	25	0	1	3	1.16	0.25
28	Definición de requisitos	23	26	0	1	2	1	0.1
29	Definición de requisitos	25	27	0	1	3	1.16	0.25
30	Definición de requisitos	x	28	1	2	4	2.16	0.25
31	Definición de requisitos	28	29	2	3	5	3.16	0.25
32	Definición de requisitos	x	30	1	2	4	2.16	0.25



El diagrama CPM con la ruta crítica no lo he podido conseguir hacer, intente de todo tipo de formas y yendo al material pero el documento en el pdf subido en CREA esta vacío, hice la tabla con los valores y no logro conseguir que me quede bien el diagrama, me mareo un montón que hayan muchas actividades que no dependan necesariamente de otra para hacerse y al momento de la representación en los nodos y actividades me quedan varias actividades en un nodo solo y no se ni por donde arrancar para calcular T1 y T2 para hallar la ruta crítica. Sin dudas en este momento hacer este CPM esta fuera de mi alcance. Dejo una foto de los tantos intentos.



11. Prototipado de la aplicación en coordinación con el profesor de FullStack

El prototipado de la aplicación no se llegó a coordinar entre los profesores y el profesor de programación descartó Muck-up y Wireframe y nos dijo que utilizáramos Figma, siendo así lo hecho en figma:

<https://www.figma.com/design/PUNSFszpfYbmDKa5xLgKOC/Sin-t%C3%ADtulo?node-id=3-275&t=11muavSOZtGFWfo1-1>



12. Carta de presentación del equipo

https://drive.google.com/file/d/1grOXmymNlic7UjPJ2cLtOCPIhd995xf4/view?usp=drive_link

13. Reglamento interno del equipo

[https://drive.google.com/file/d/1O8MMVFr4tLZy-5nfqa6U4p2CQ42w_Fwf/view?usp=drive link](https://drive.google.com/file/d/1O8MMVFr4tLZy-5nfqa6U4p2CQ42w_Fwf/view?usp=drive_link)

14. Creación Repositorio privado en Git y subir

El repositorio ya fue creado y ya se encuentra TODO el proyecto subido al repositorio como se puede ser en la captura

