

PROYECTO INFOBICIS

ALUMNOS (INTEGRANTES):

*BANDA MORALES ROBERTO
NICOLÁS SANTIAGO JOSÉ FERNANDO*

PROFESOR:

VÁZQUEZ SANTACRUZ EDUARDO

UEA:

TEMAS SELECTOS DE INGENIERÍA DE SOFTWARE

GRUPO:

CJ01

PROBLEMÁTICA

En la actualidad el uso de la bicicleta se ha hecho más común entre las personas en todo México, cada vez más personas son las que utilizan este transporte para ir a su trabajo, escuela o alguna actividad recreativa. Según datos del instituto nacional de estadísticas y geografía (INEGI), en la Ciudad de México y área metropolitana hay aproximadamente dos millones cien mil bicicletas. Esto ha generado que la delincuencia se adapte a nuevas formas de robo ya que hay bicicletas que alcanzan precios extremadamente altos y la demanda de comprar por una bicicleta a un buen precio va en aumento, por lo tanto, el robo de bicicleta es un tema recurrente para los habitantes en la Ciudad de México. Se considera que al menos más de 6 bicicletas son robadas al día solo en la Ciudad de México Y después son vendidas en tianguis, mercado libre, segunda mano y grupos de venta en Facebook.

Muchas veces cuándo se compra una bicicleta usada, no se cuenta con los papeles o algún documento que acredite que la bicicleta no es robada por lo tanto hay muchas posibilidades de que adquieras una bicicleta robada y esto te pueda causar algún problema legal, existen grupos de Facebook de robo de bicicletas, donde tu subes una publicación con foto de tu bicicleta, explicando que fue robada, pero no son muy útiles, por que suben tantas publicaciones de robos de bicicletas que al final del día tu publicación queda enterrada entre tantas publicaciones.

PROPUESTA DE PROYECTO

Este proyecto se propone crear una plataforma para la verificación de bicicletas si tienen reporte de robo o no, en dado caso de que estén vendiendo una bicicleta que tenga reporte de robo, puedes ponerte en contacto con el afectado para informarle de la situación, con esto se espera aminorar el problema de cuando se adquiere una bicicleta que es robada.

LISTA DE ACTIVIDADES

ID ↑	ASUNTO	TIPO	ESTADO	ASIGNADO A	PRIORIDAD	*
34	Análisis del problema	TASK	New	RB Roberto Banda	High	
35	Levantamiento de Requerimientos	TASK	New	FN Fernando Nicolás	Immediate	
36	Documentación y Análisis de CU	TASK	New	FN Fernando Nicolás	Immediate	
37	Documentación y Análisis de CU	TASK	New	RB Roberto Banda	Immediate	
38	Diseño de BD	TASK	New	RB Roberto Banda	High	
39	Diseño de interfaz	TASK	New	FN Fernando Nicolás	High	
40	Diseño de Software	TASK	New	FN Fernando Nicolás	High	
41	Desarrollo del Sistema	TASK	New	RB Roberto Banda	High	
42	Integrar módulos	TASK	New	FN Fernando Nicolás	High	
43	Realizar pruebas	TASK	New	RB Roberto Banda	Normal	
+ Crear un nuevo paquete de trabajo						

DIAGRAMA DE GANTT

Diagrama de Gantt

+ Crear

Filtro 1

Gantt

?

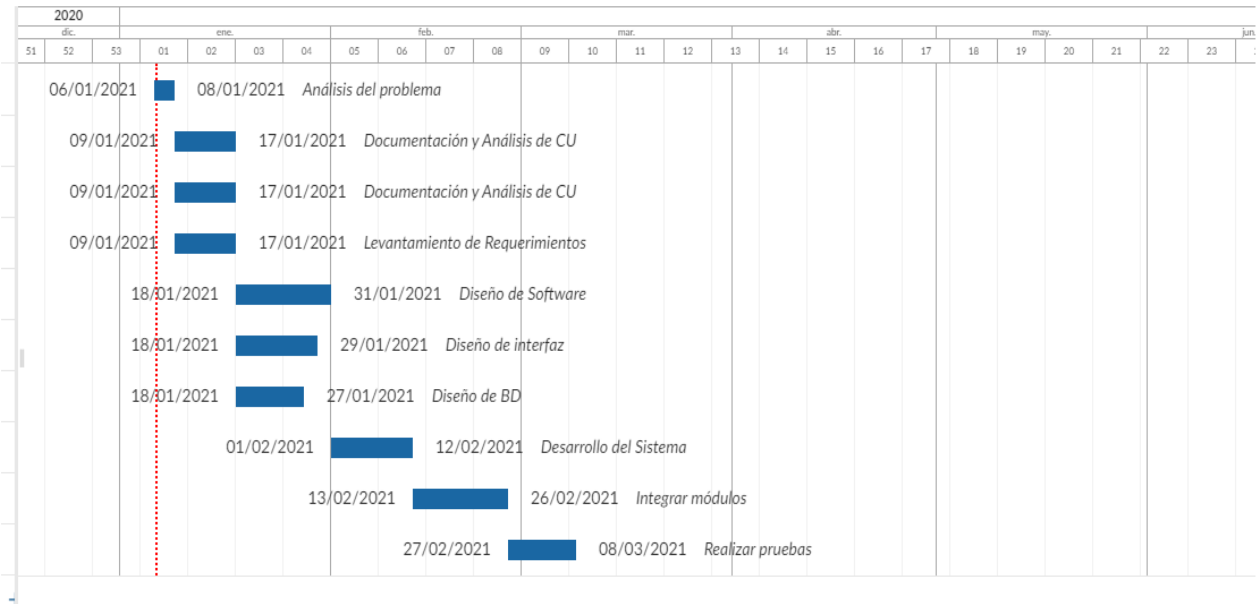
🔍

🔍

🔍

🔍

⋮



COSTOS DEL PROYECTO

Integrantes	Especialidad	Sueldo Mensual	Tiempo 3 Meses
Fernando Nicolás	Líder de proyecto	\$ 13,000.00	\$ 39,000.00
Roberto Banda	Programador	\$ 10,000.00	\$ 30,000.00

Costos Operativos	\$ 8,000.00
Costos Personal	\$ 69,000.00
Subtotal	\$ 77,000.00
Riesgo (30%)	\$ 23,100.00
Total Proyecto	\$ 100,100.00

PROCESO DE DESARROLLO DE SOFTWARE: MODELO DE MADUREZ DE CAPACIDADES INTEGRADO (CMMI)

En nuestro proyecto a desarrollar escogemos este modelo para el proceso de desarrollo de software, el cual implica las siguientes etapas a llegar al finalizar nuestra aplicación:

a. Niveles de Madurez por Etapas:

- I. **Inicial.** El proceso es impredecible, es reactivo y poco controlado.
- II. **Administrado.** El proceso es reactivo y se caracteriza por su aplicación a proyectos.
- III. **Definido.** El proceso es proactivo y se ve a nivel de organización.
- IV. **Administrado Cuantitativamente.** El proceso es medido y controlado.

ANÁLISIS DE FACTIBILIDAD

Bloque/Tipo de riesgo	Recursos agotados	Conocimientos	Planificación	Integración
Análisis de Requerimientos	Se realiza sin muchos costos.	Se realiza con la complicación de que no se capture lo que el cliente pide.	Requiere tiempo para realizar el análisis.	No requiere modificaciones de dos personas, solo el analista.
Diseño de BD	No necesita muchos costos para el diseño de la BD.	Se puede tener errores de diseño de la BD lo cual impactaría en el sistema.	La mala planificación de la BD afectaría en el sistema	No requiere que muchas personas para la integración, solo el analista de BD.
Diseño de Interfaz	Se realiza con medianos costos de diseño.	El no tener el conocimiento suficiente retrasaría el diseño de la interfaz.	El no tener tiempo suficiente retrasaría el incluir las demás funciones del sistema.	Podría repercutir la mala integración de los developers.
Desarrollo del Sistema	Se realiza con altos costos de desarrollo del sistema.	Si se tiene dudas con el documento de requerimientos, puede repercutir cuando se desarrolle el sistema.	Puede tener errores en las funciones que se implementan.	Mala integración de los developers puede influir en el sistema, en caso contrario puede funcionar la buena integración.
Pruebas del Sistema	Se realiza con altos costos en las pruebas.	Se realizan pruebas unitarias.	Se cuenta con menos tiempo para la realización de las pruebas.	No se solucionan los posibles errores.
Entrega	Entrega cumpliendo los requerimientos.	Entrega completa	Entrega con toda la planificación normal.	Entrega con las integraciones sin errores.

PLAN DE PRUEBAS

Dentro de nuestra aplicación se llevarán ciertas técnicas para la realización de pruebas, pues son las que se utilizan para el diseño de los casos de prueba.

PROPUESTA DE TÉCNICA DE DISEÑO DE PRUEBAS

El modelo que elegimos para nuestro proyecto es el método de prueba de caja negra, dicha técnica consiste en:

1. Verificar el correcto manejo de funciones externas soportadas por el software.
2. Verifica que el comportamiento se apegue a las especificaciones del usuario, así como la satisfacción de sus expectativas.
3. Los casos de prueba se construyen a partir de los requerimientos del sistema.

Esta técnica permite derivar casos de prueba que buscan solucionar los siguientes errores:

- ✓ Funciones incorrectas.
- ✓ Errores de interfaz.
- ✓ Errores en estructuras de datos o a base de datos externas y/o internas.
- ✓ Errores de comportamiento o desempeño.

ESTRATEGIA DE PRUEBAS

Se utilizará el modelo en V para realizar de manera eficiente los distintos niveles de prueba:

1. *Pruebas unitarias.* Verificaremos las unidades del software de forma aislada, es decir, se prueba el buen funcionamiento del código.
2. *Pruebas de integración.* Verificaremos que, al juntar las partes o componentes del proyecto, no ocurran errores que no se tienen previstos.
3. *Pruebas de sistema.* Verificaremos que el sistema cumpla con los requisitos especificados por el cliente.
4. *Pruebas de aceptación.* El cliente verificará el correcto funcionamiento del sistema.

CRITERIOS DE ENTRADA, SALIDA, SUSPENSIÓN DE PRUEBAS

Entrada:

- a. Aprobación del código que pasa por el ciclo de pruebas.
- b. Ejecución de pruebas unitarias realizadas por el área de desarrollo.
- c. Aprobación del líder del proyecto.

Salida:

- a. Pruebas unitarias.
 - ✓ Detecta fallos en la ejecución de pruebas.
 - ✓ El resultado de la ejecución debe ser exitosa.
- b. Pruebas de sistema.
 - ✓ Detecta fallos en la ejecución de pruebas.
 - ✓ El resultado de cada caso de prueba debe ser exitosa.
 - ✓ El sistema debe cumplir con el flujo previsto.
- c. Pruebas de aceptación.
 - ✓ Para la realización de esta prueba debe respetar los requerimientos asociados al desarrollo de software.
 - ✓ Debe cumplir con el 100% de pruebas sin errores.

Suspensión:

- a. El área de desarrollo no realiza las pruebas correspondientes.
- b. Se presentan defectos que impidan el funcionamiento del sistema.

ANÁLISIS DE RIESGO

Tipo de Riesgo	Descripción	Probabilidad	Efecto
Tecnológico	✓ La base de datos usada para la plataforma no puede procesar tantas búsquedas cómo se había planificado.	Baja	Tolerable
Organización	✓ Los problemas financieros de la organización llevan a reducir el personal asignado al proyecto.	Mediana	Tolerable
Técnicos	✓ No todo el personal conoce el lenguaje JAVA con el Framework Spring Boot.	Alta	Seria
Requerimientos	✓ Los cambios en los requerimientos obligan hacer cambios en el diseño de la plataforma. ✓ Algunos requerimientos no son cómo el cliente propuso.	Alta	Catastrófica
Estimación	✓ El tiempo de pruebas de software no se estimo adecuadamente. ✓ El costo del personal no se considero adecuadamente.	Alta	Seria

CRONOGRAMA

