

Entrega 1

Proyecto

Samuel García - Ing. Civil  
Febe Oropesa - Ing Ambiental

Lógica computacional  
Profesor : Juan Martínez.

ESTILO

## Actividades preliminares

Contexto ④ Conocer el ciclo de vida de un proyecto nos permite controlar mejor sus fases y reducir situaciones innecesarias. En cualquier proyecto, las fases son las mismas, aunque cada proyecto requiere su propio conjunto único de procesos. Se compone de varias etapas clave.

1. Requisitos: Recopilación y análisis de las necesidades del usuario.
2. Diseño: Planificación de la arquitectura del software y definición de la interfaz.
3. Implementación: Programación y codificación del software.
4. Pruebas: Verificación de que el software funcione correctamente y cumpla todos los requisitos.
5. Despliegue: Instalación y puesta en marcha del software en el entorno del cliente.
6. Mantenimiento: Corrección de errores y actualizaciones después del despliegue.



Existen modelos que nos permiten establecer una metodología que nos servirá de guía para validar todas las etapas del proyecto.

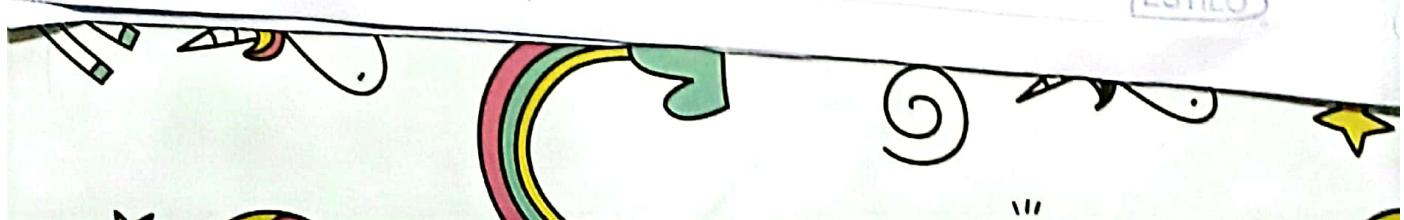
Modelo en cascada: es un método de gestión que divide un proyecto en distintas fases secuenciales. Todas ellas funcionan de forma lineal, es decir, que cada parte del proyecto se completa antes de empezar con la siguiente. Este modelo es simple y fácil de usar, ya que todo está bien organizado y las fases no pueden mezclarse entre sí.

Modelo en V: Se basa en la gestión de proyectos de forma lineal, pero consta de una fase descendente y otra ascendente.

fase descendente: tareas de diseño y desarrollo  
fase ascendente: control de calidad.

### b) Aspectos del análisis de un problema

1. Definición del problema: Escriba una frase que resuma el problema que se analiza en el centro de una hoja de papel. Comprender lo que se debe resolver
2. Recolección de datos: obtener información relevantes sobre el problema.
3. Identificación de causas: determinar las causas raíz del problema.
4. Evaluación y selección de la solución: poner en práctica la solución elegida. Esto incluye la planificación de los pasos necesarios y asignación de recursos
5. Evaluación de los resultados: Una vez implementada la solución, es importante monitorear y evaluar su efectividad. Si la solución no funciona como se esperaba, puede ser necesario volver a etapas para ajustar y cambiar el enfoque.



### C. Explique las etapas del proceso de solución de problemas

La resolución de problemas es un método sistemático para solucionar un problema. El objetivo de resolución de problemas es determinar por qué algo no funciona como se esperaba y cómo solucionar el problema. Puede Buscar recursos de soporte activos basados en la Web utilizando el formulario de búsqueda web situado bajo búsqueda en bases de conocimiento en el panel de navegación de la izquierda.

El primer paso del proceso de resolución de problemas es describir por completo el problema ayudan al usuario y al representante del soporte de IBM a saber por dónde comenzar a buscar la causa del problema. Este paso incluye formularse las siguientes preguntas básicas:

- ¿Cuáles son los síntomas del problema?
- ¿Dónde se produce el problema?
- ¿Cuándo se produce el problema?
- ¿En qué condiciones se produce el problema?
- ¿El problema puede reproducirse?

#### ¿Dónde se produce el problema?

Determinar dónde se origina el problema no siempre es fácil, pero es una de los pasos más importantes para resolver un problema. Pueden existir muchas capas de tecnología entre el componente con el error y el componente que lo comunica. Redes, discos y controladores son solo algunos de los componentes a tener en cuenta cuando se investiguen problemas.

Las preguntas ayudan a resolver los problemas.

#### ¿Cuándo se produce el problema?

Desarrolle una secuencia detallada de los eventos que conducen a un error, especialmente en los casos que se producen una sola vez. La manera más fácil de desarrollar una línea temporal es ir hacia atrás a través de los registros y la información disponibles. Normalmente solo es necesario buscar hasta el primer suceso.

#### ¿En qué condiciones se produce el problema?

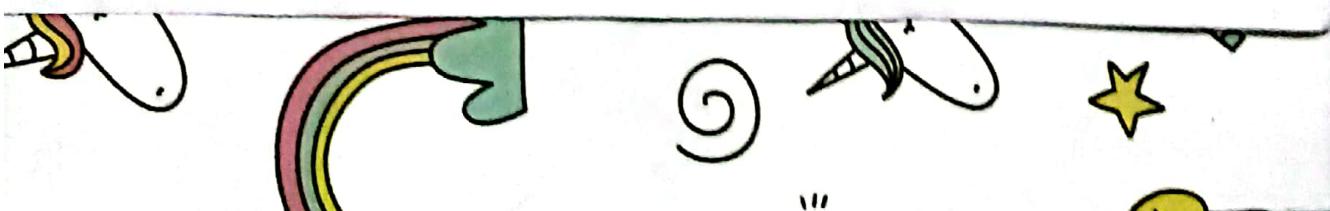
Saber qué sistemas y aplicaciones se están ejecutando en el momento en que se produce el problema es una parte importante de la resolución de problemas. Estas preguntas sobre el entorno pueden ayudar a identificar el problema.

#### ¿El problema siempre se produce al realizar la misma tarea?

#### ¿El problema puede reproducirse?

Un problema ideal es aquel que puede reproducirse. Normalmente cuando un problema se puede reproducir, se dispone de un conjunto de herramientas o procedimientos mayor para ayudar a investigarlos. Por lo tanto, los problemas que se pueden reproducir suelen ser más fáciles de depurar y resolver. Sin embargo, los problemas que pueden reproducir pueden tener un impacto empresarial significativo, no querrá que se repita.

ESTRUCTURA





d) Elementos que se deben entregar a un cliente:

1. Código fuente: El código que implementa la solución. Debe estar bien documentado y organizado.
2. Documentación técnica: Instrucciones sobre cómo instalar, configurar y utilizar el software. También puede incluir diagramas y explicaciones de la arquitectura del programa.
3. Manual de usuario: Un documento que guía al usuario final sobre cómo utilizar la aplicación de manera efectiva.
4. Pruebas: Resultados de las pruebas realizadas, incluyendo casos de pruebas, informes de errores y aceptación del usuario (UAT).
5. Plan de mantenimiento: Instrucciones sobre cómo mantener y actualizar el software, así como el soporte disponible.
6. Licencias y derechos de uso: Información sobre los derechos de uso del software y cualquier restricción aplicable.
7. Capacitación /reservas necesarias: Si el software es complejo, puede ser útil ofrecer capacitación para los usuarios.

e) Elabore la tarea No 1 (pág 5 del texto guía), con el objetivo de identificar los aspectos que forma parte de un problema.

Cliente: El Banco o institución financiera que quiere implementar el sistema de cajeros automáticos

usuario: Clientes del banco que utilizan los cajeros automáticos

Requerimiento funcional:

- El programa debe permitir a los usuarios realizar retiros de dinero.
- El programa debe permitir a los usuarios consultar el saldo de su cuenta.
- El programa debe validar la autenticación del usuario (mediante tarjeta y Pin).

Mundo del problema: El sistema debe estar conectado a la base de datos del banco, donde se almacenan los saldos y transacciones de los usuarios, debe garantizar que haya fondos suficientes.

Requerimiento funcional: El programa debe ser seguro para evitar fraudes, además, el sistema debe ser rápido eficiente, permitiendo que las transacciones se completen en un tiempo razonable.

ESTILO



7) Elabore la tarea No 2. (Pág 13), con el objetivo de identificar los requerimientos de un problema.

Requerimiento funcional 1

Nombre: Consultar saldo de cuenta  
Resumen: Permite al usuario consultar el saldo actual de su cuenta bancaria en el simulador.  
Entradas: Número de cuenta  
Tipo de cuenta (ahorro, corriente)  
Resultado: Muestra el saldo disponible en la cuenta solicitada.

Requerimiento funcional 2.

Nombre: Realizar transferencias entre cuentas  
Resumen: Permite al usuario realizar una transferencia de fondos entre dos cuentas dentro del simulador bancario.  
Entradas: Número de cuenta de origen, Número de cuenta destino, Monto a transferir, fecha de la transferencia.  
Resultado: Transacción completada y actualización de saldo de las cuentas de origen y destino.

Requerimiento funcional 3

Nombre: Simulación de pago de préstamo.  
Resumen: Permite al usuario realizar un pago de su préstamo dentro del simulador reduciendo el saldo pendiente.  
Entradas: Número de cuenta del préstamo, monto a pagar, fecha de pago.  
Resultado: El saldo del préstamo disminuye según el monto pagado, y actualiza el estado del usuario.

8) Elabore tarea N° 3 (pág 14).

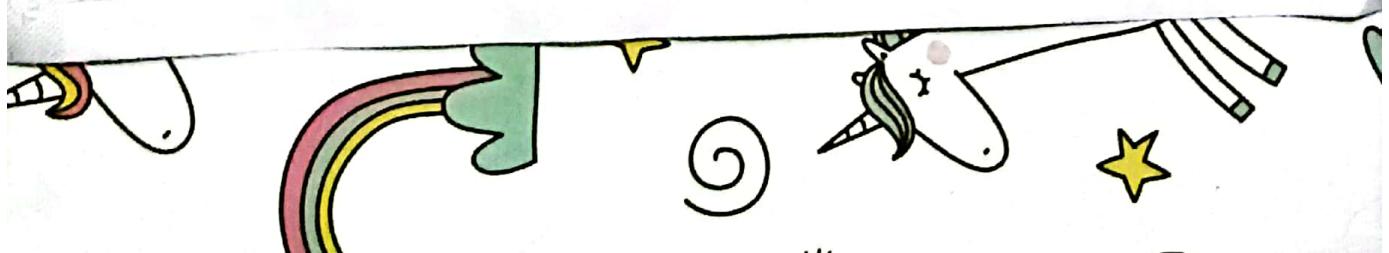
Requerimiento funcional 1

Nombre: Ingresar las medidas de un triángulo  
Resumen: Permite al usuario ingresar las longitudes de los tres lados del triángulo.  
Entradas: Longitud del lado A, del lado B, lado C.  
Resultado: Medidas de los lados del triángulo registradas

Requerimiento funcional 2.

Nombre: Calcular el perímetro  
Resumen: Con las longitudes de los 3 lados se calculará el perímetro.  
Entradas: Longitud A, B, C.  
Resultado: Perímetro del triángulo.

ESTILO



Requerimiento funcional 3.

Nombre:

Determinar el tipo de triángulo.

Resumen: Con las medidas de los lados, el sistema identificará el tipo de triángulo.

Entradas: Longitud A, B, C.

H.

1. Triángulo

Descripción: la figura geométrica que se va a manejar en el programa. Puede tener propiedades como lados y ángulos.

2. Lado.

Un atributo de la entidad triángulo, que corresponde a la longitud de los tres lados.

3. Ángulo.

Otro atributo del triángulo, que define los tres ángulos internos de la figura.

1. ¿Qué pasa si no identificamos las entidades del mundo?

El programa podría estar mal diseñado y no cumplir con los requerimientos funcionales, lo que llevaría a errores en la lógica del programat o una implementación incompleta.

2. ¿Cómo decidir si se trata efectivamente de una entidad y no sólo de una característica de una entidad y a identificada?

Es importante preguntarse si ese elemento puede existir por sí mismo dentro del contexto del problema o si depende de otra entidad para tener sentido. Si depende de otra entidad su existencia, probablemente sea una característica o atributo.

ESTILO



II

## Clase: Cuenta Bancaria

## Atributos:

- Número de cuenta

Valores posibles  
cadena de  
caracteres

Comentario  
Identificador único de la  
cuenta, necesario para  
realizar transacciones

saldo

decimal  
positivo

Monto actual disponible  
en la cuenta,  
que puede varcar

tipo

Enumeración

Especifica el tipo  
de cuenta

## Clase corriente

## Atributos

- número de  
cuenta

Valores posibles

cadena de  
caracteres

## Comentario

Identificador único  
para la cuenta  
corriente.

saldo

Decimal  
positivo

Monto disponible en  
la cuenta corriente,  
puede incluir  
sobregiro

sobre giro

decimal

Monto maximo que el  
cliente puede acceder

## Clase 3. Cuenta Ahorros

## Atributos

- Número de  
cuenta

Valores posibles

cadena de  
caracteres

## Comentario

Identificador único de  
la cuenta de  
ahorros.

Saldo

decimal positivo

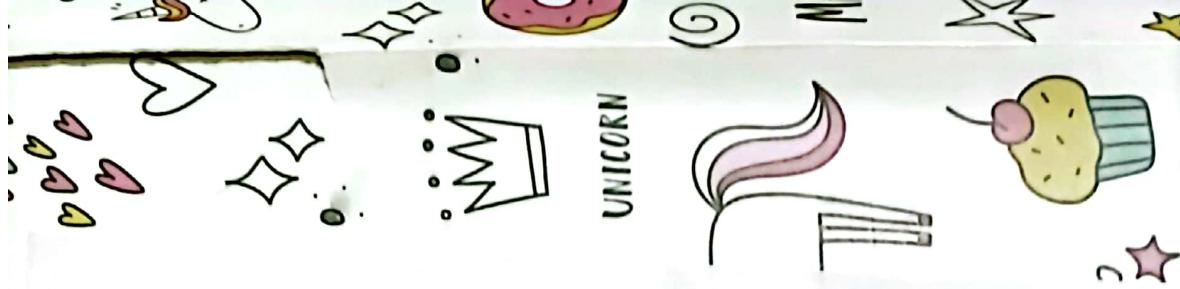
Monto disponible en la  
cuenta.

Interés

decimal

tasa de interes  
anual aplicada al  
saldo

ESTILO



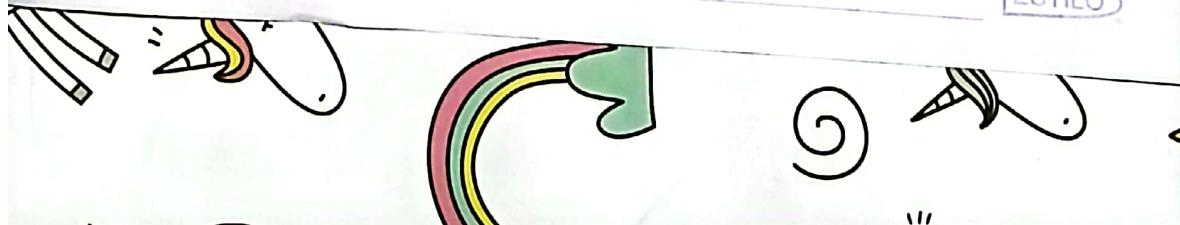
### Clase 4 cliente.

Atributo	valores posibles	comentario
Nombre	Cadena de caracteres	Nombre completo del cliente, utilizando en la identificación.
documento de identidad	Cadena de caracteres	
telefono	Cadena de caracteres	Número de identificación
correo electrónico	Cadena de caracteres	Número de contacto del cliente.
		Dirección de correo electrónico

### Clase 5 transacción

Atributo	valores posibles	comentario
id transacción	Cadena de caracteres	Identificador único de transacción
fecha	fecha	fecha en que se realizó la transacción
Monto	Decimal Positivo	Monto de la transacción
Tipo	Enumeración (Depósito, Retiro, Transferencia)	Tipo de operación realizada.

ESTILO



J. Elabore la tarea N° 6 (pág 23).

1. Compra de tiquetes: No es especifica el tipo de tiquete ni el método de pago.
2. Mapa del metro: la interpretación del mapa puede ser confusa para alguien no familiarizado.
3. Nombre de la estación: Puede haber múltiples estaciones cercanas sin claridad de saber cuál es correcta.
4. Verificación de líneas: No se menciona qué hacer si no hay línea directa o si se necesitan transbordos.
5. Nombres de estaciones: Podrían existir nombres similares que confundan al usuario.
6. Subir al metro: No se indica claramente cómo identificar el andén o dirección correcta.

Un algoritmo que busca ser funcional debe evitar ambigüedades y ser lo suficientemente detallado para cualquier persona, sin ningún conocimiento previo, para que pueda seguirlo correctamente. En este caso, las instrucciones asumen que el usuario tiene un nivel de conocimiento sobre el sistema de transporte o un "sentido común" sobre cómo actuar, pero esto no siempre es el caso.

- Etiquetar el tipo de tiquete adecuado
- Induir una guía visual o como interpretar un mapa.
- Explicar cómo identificar correctamente los andenes y verificar el sentido del tren.

K. Estudie los siguientes aspectos del ejemplo seleccionado: Enunciado, requerimiento funcionales (casos de uso) y modelo (fase del proyecto).

### Enunciado . . . triangulo

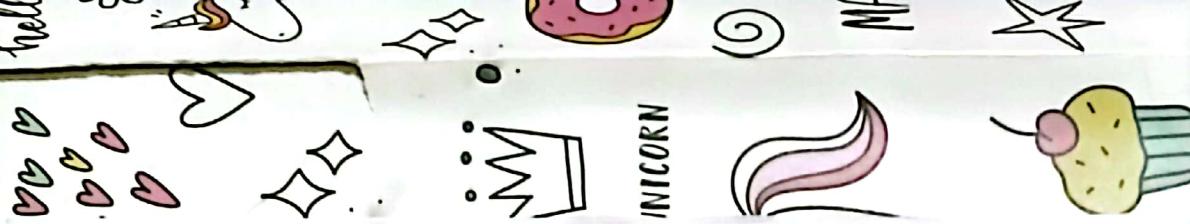
Se quiere construir un programa que permita dibujar un triángulo. Esta figura geométrica está definida por tres puntos, cada uno de los cuales tiene dos coordenadas x, y. Este triángulo tiene, además un color para las líneas y un color para el relleno.

Un color, por su parte, está definido por tres valores numéricos entre 0 y 255 (estándar RGB por Red-Green-Blue). El primer valor numérico define la intensidad en Rojo, el segundo en verde y el tercero en Azul.

El programa debe permitir:

1. Dibujar el triángulo en la interfaz.
2. Cambiar los puntos del triángulo.
3. Cambiar el color de relleno del triángulo.
4. Cambiar el color de las líneas.
5. Mostrar perímetro.
6. El Área del triángulo.
7. Altura del triángulo.

ESTILO

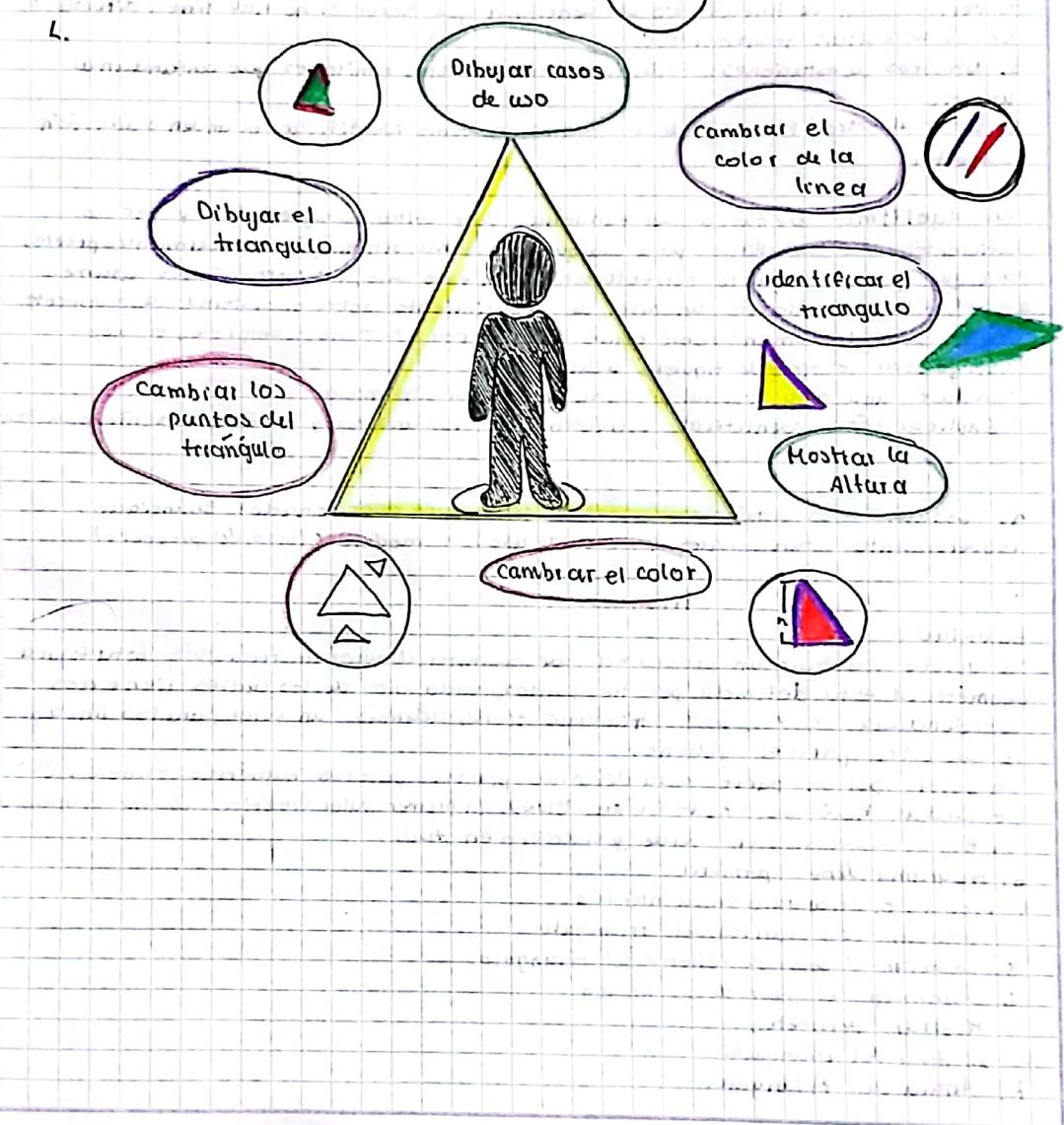


Nombre de requerimientos:

- Dibujar el triángulo en la interfaz
- Cambiar los puntos del triángulo
- Cambiar color de relleno del triángulo
- Cambiar color de las líneas del triángulo
- Mostrar el área del triángulo
- Mostrar la altura del triángulo
- Mostrar el perímetro del triángulo

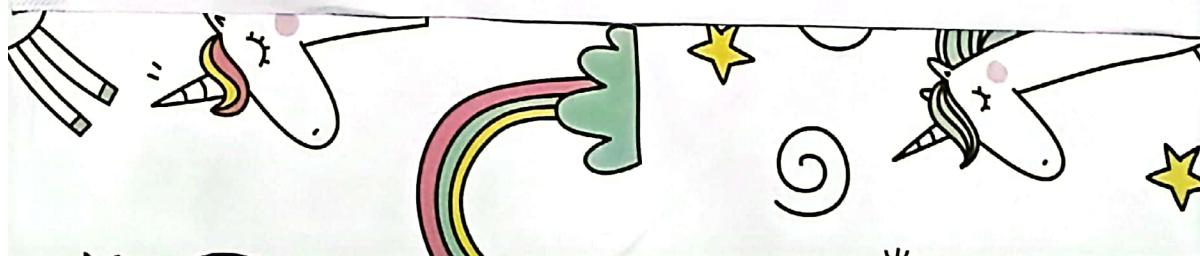
UNICORN

L.



Tu mundo. Tu estilo. Tu magia.

ESTILO



n. Debes plantear 2 ideas de proyecto (problemas solubles y algorítmicos).

ideal: Aplicación de Gestión de tareas

Requerimiento funcional 1:

Nombre: Crear tarea

Resumen: Permite a los usuarios añadir nuevas tareas a su lista.

Entrada: título de la tarea, descripción, fecha de vencimiento, prioridad

Resultado: La tarea se agrega a la lista de pendientes.

Requerimiento funcional 2:

Nombre: Ver lista de tareas

Resumen: Muestra todas las tareas agregadas con su estado (pendiente/completa).

Entradas: Ninguna.

Resultado: Una lista que muestra todas las tareas con detalles relevantes.

Requerimiento funcional 3:

Nombre: Actualizar tarea

Resumen: Permite a los usuarios eliminar una tarea de la lista.

Entradas: ID de la tarea, nuevo título, nueva descripción, nueva fecha, nueva prioridad.

Requerimiento funcional 4:

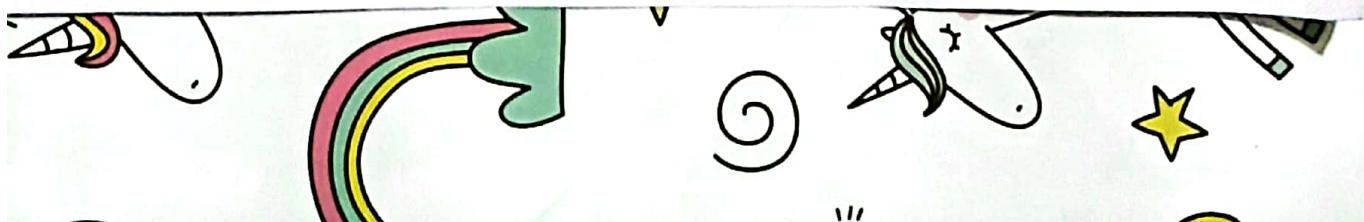
Nombre: Eliminar tarea

Resumen: Permite a los usuarios eliminar una tarea de la lista.

Entradas: ID de la tarea

Resultado: La tarea seleccionada se elimina de la lista de tareas.

ESTILO



## Idea 2. Sistema de Reserva para un Restaurante.

Requerimiento funcional 1

Nombre: Hacer reserva

Resumen: Permite a los clientes reservar el restaurante en una mesa.

Entradas: Nombre del cliente, fecha, hora, número de personas.

Resultado: se confirma la reserva y se envía un correo de confirmación al cliente.

Requerimiento funcional 2

Nombre: Consultar Disponibilidad

Resumen: Permite a los usuarios verificar la disponibilidad de mesa en una fecha y horas específicas.

Entradas: fecha, hora, número de personas.

Resultado: indica si hay mesas disponibles.

Requerimiento funcional 3

Nombre: Modificar Reserva

Resumen: Permite a los clientes cambiar los detalles de una reserva.

Entradas: ID de la reserva, nueva fecha, nueva hora, nuevo número de personas.

Resultado: La reserva se actualiza.

Requerimiento funcional 4.

Nombre: Cancelar reserva

Resumen: Permite a los clientes cancelar una reserva existente.

Entradas: ID de la reserva

Resultado: La reserva se elimina del sistema.

ESTILO