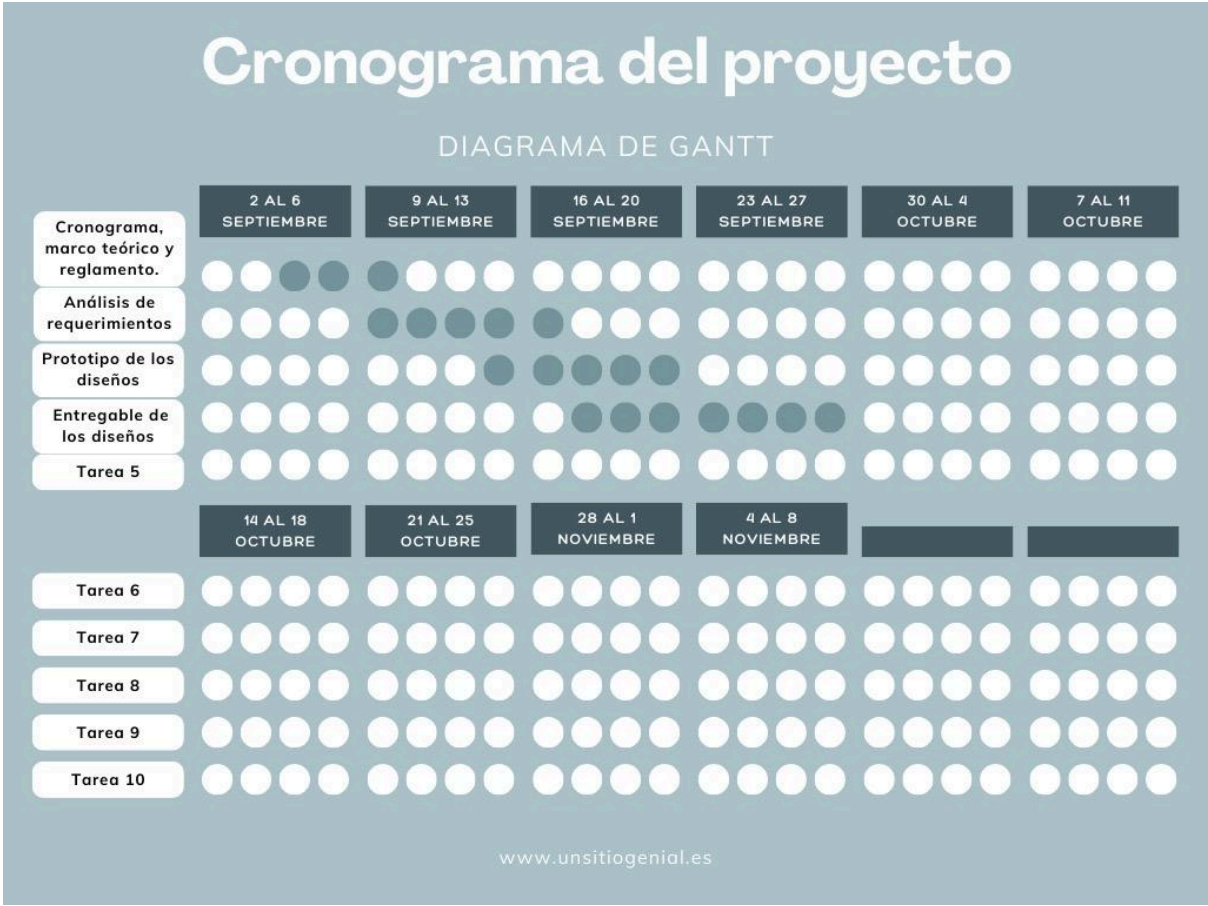


PRÁCTICA 0.

Índice.

Índice.	2
Cronograma.	3
Marco teórico.	4
MySQL	4
MySQL Cluster	4
Modelos	4
Palabras reservadas de MySQL	4
UML	5
CASO DE USO	5
JAVA	6
INTERFACES GRÁFICAS.	6
CLASES	6
VARIABLES.	6
OPERADORES	7

Cronograma.



Marco teórico.

MySQL

Para poder cumplir con la creación de un sistema eficiente y de calidad usaremos MySQL como sistema de gestión de bases de datos relacionales. Lo anterior debido a la experiencia que se tiene acerca de ese RDBMS, además de robustez, capacidad de manejo de grandes volúmenes de datos, sin olvidar su fácil integración con distintos lenguajes de programación. Una base de datos distribuida se compone de múltiples nodos interconectados, cada uno de los cuales almacena una parte de la información total. Algunos conceptos clave son:

- **Fragmentación:** Dividir la base de datos en fragmentos más pequeños para distribuirlos entre los nodos. Puede ser horizontal (por filas) o vertical (por columnas).
- **Replicación:** Mantener copias idénticas de los datos en varios nodos para mejorar la disponibilidad y la tolerancia a fallos.
- **Transacciones distribuidas:** Coordinar operaciones que afectan a múltiples nodos para mantener la consistencia.

MySQL Cluster

MySQL Cluster es una solución de base de datos distribuida proporcionada por MySQL.

Algunos aspectos relevantes son:

- **Nodos:** Un clúster MySQL consta de varios nodos, incluyendo nodos de datos, nodos de gestión y nodos SQL.
- **Almacenamiento distribuido:** Los datos se distribuyen entre los nodos de datos para lograr alta disponibilidad y escalabilidad.
- **Replicación síncrona:** MySQL Cluster admite replicación síncrona para garantizar la consistencia de los datos.

Modelos

- **Modelo Relacional:** El modelo relacional es el más común en bases de datos. Se basa en tablas (relaciones) con filas y columnas. Cada tabla representa una entidad (por ejemplo, Clientes, Instructores, Cursos) y las relaciones entre ellas se establecen mediante claves primarias y foráneas.
- **Modelo Entidad-Relación (ER):** Este modelo se utiliza para diseñar la estructura lógica de la base de datos. Representa entidades, atributos y relaciones entre ellas mediante diagramas ER.
- **Modelo Físico:** El modelo físico describe cómo se implementa la base de datos en el sistema de gestión de bases de datos (DBMS). Incluye detalles como índices, claves y restricciones.

Palabras reservadas de MySQL

- **SELECT:** Recupera datos de una o más tablas. Es la base de las consultas y permite especificar qué columnas se desean ver y bajo qué condiciones.
- **INSERT INTO:** Agrega filas a una tabla. Se utiliza para insertar nuevos registros en la base de datos.
- **UPDATE:** Modifica los valores de una fila existente en una tabla. Útil para actualizar información.
- **DELETE FROM:** Elimina filas de una tabla. Permite borrar registros.
- **ALTER TABLE:** Modifica la estructura de una tabla existente, como agregar o eliminar columnas.
- **CREATE TABLE:** Crea una nueva tabla con sus columnas y restricciones.
- **JOIN:** Combina datos de dos o más tablas basándose en una condición. Por ejemplo, para obtener información de clientes junto con sus instructores.
- **GROUP BY:** Agrupa filas según una columna específica. Útil para cálculos agregados como contar o promediar.
- **ORDER BY:** Ordena los resultados según una o más columnas. Puede ser ascendente (por defecto) ó descendente.
- Funciones de agregación: Incluyen **SUM**, **COUNT**, **AVG**, **MIN** y **MAX**. Realizan cálculos sobre grupos de filas.

UML

Se utilizarán diagramas UML para la modelación visual del sistema, lo cual permitirá representar de manera clara la estructura y las interacciones entre los diferentes componentes del sistema.

CASO DE USO

Un diagrama de casos de uso nos permitirá identificar y visualizar todas las funcionalidades que el sistema debe ofrecer a los diferentes tipos de usuarios (actores). Además de que es visualmente sencilla de comprender, esto facilita la comunicación entre desarrolladores, clientes, y otros involucrados en el proyecto.

Teniendo en cuenta lo anterior hay que definir los puntos claves para la buena implementación del diagrama de casos de uso:

1. Actores.
2. Descripción
3. Precondiciones.
4. Postcondiciones.
5. Flujo normal.
6. Flujos alternativos.
7. Excepciones.
8. Prioridades
9. Frecuencia de uso.
10. Reglas.
11. Requerimientos especiales.

12. Suposiciones.

Estos diagramas no especifican el comportamiento de los casos de uso, sino solamente relaciones entre los distintos casos de uso, y casos de uso con actores.

JAVA

La interfaz de usuario será desarrollada en Java, lo que permitirá ofrecer una experiencia interactiva y amigable para los administradores del sistema. Además, el uso de Java facilita la escalabilidad y mantenimiento del software, garantizando que la aplicación pueda adaptarse al crecimiento de la franquicia.

INTERFACES GRÁFICAS.

Una gran ventaja de trabajar con java, es que se puede trabajar con ventanas o interfaces gráficas. Y se comportan de la misma forma independientemente del sistema operativo con el que trabajemos (windows, linux etc) y, en muchas ocasiones, utilizan los mismos componentes básicos para introducir órdenes (menús, botones, cuadros de texto, etc.).

Por ejemplo, una ventana típica de windows tiene las siguientes partes:

1. barra de menús.
2. Icono de aplicación y menú de control.
3. barra de título.
4. Botón para minimizar la ventana.
5. Botón para maximizar la ventana.
6. Botón para cerrar la ventana.
7. barra de desplazamiento vertical.
8. Marco de la ventana.
9. Barra de desplazamiento horizontal.
10. Área de trabajo.

Las interfaces gráficas se usan para interaccionar con el usuario, mostrando todas las opciones que el usuario puede realizar. Por lo tanto el diseño consistirá en crear objetos que den lugar a ventanas y sobre esas ventanas se dibujaran otros objetos llamados controles. Cada objeto estará ligado a un código que estará inactivo hasta que se produzca el evento que lo activa.

CLASES

El lenguaje java es un lenguaje orientado a objetos. La base de la POO (Programación orientada a objetos) es la clase. Una clase es un tipo de objeto definido por el usuario. Una característica que aporta la POO es la herencia ya que permite la reutilización del código escrito por nosotros o otros.

VARIABLES.

Una variable representa un espacio de memoria para almacenar un valor de un determinado tipo. La declaración de una variable puede realizarse en el ámbito de la clase (dentro de la clase pero fuera de todo método), en el ámbito del método o en el ámbito de un bloque cualquiera delimitada por { }.

OPERADORES

Los operadores son símbolos que indican cómo los datos se manipulan. Se pueden clasificar en los siguientes grupos:

- ❖ Aritméticos:
 - + Suma. Los operandos pueden ser enteros o reales.
 - Resta. Los operandos pueden ser enteros o reales.
 - * Multiplicación. Los operandos pueden ser enteros o reales
 - / División. Los operandos pueden ser enteros o reales. Si ambos operandos son enteros el resultado es entero. En el resto de los casos el resultado es real.
 - % Resto de una división. Los operandos pueden ser enteros o reales. Si ambos operandos son enteros el resto será entero, en otro caso, el resto será real.
- ❖ Relación:
 - < Menor que.
 - > Mayor que.
 - <= Menor igual que.
 - >= Mayor igual que.
 - != Distinto que.
 - == Idéntico que.
- ❖ Lógicos:

El resultado de una operación lógica (AND, OR, XOR Y NOT) es un valor booleano verdadero o falso (true o false) y estos son los operadores:

 - ❖ && o & AND
 - ❖ || o | OR
 - ❖ ! NOT
 - ❖ ^ XOR

Referencias

- López, C. P. (2008). MySQL para Windows y Linux.
- Wiederhold, G. (1985). Diseño de bases de datos.
- Fontela, C. (2012). UML modelo de software para profesionales.
- Shaver, D. (1998). El siguiente paso en la mercadotecnia directa: cómo usar bases de datos orientándolas al consumidor.
- Hernandez. (2010). METODOLOGIA DE LA INVESTIGACION.
- De Miguel Castaño, A., Fernandez, P. M., & Galan, E. C. (2001). Diseño de bases de datos: problemas resueltos.
- Cuadra, D., Castro, E., & Iglesias, A. M. (2013). Desarrollo de bases de datos: casos prácticos desde el análisis a la implementación.