**1:**

**a. Clase:** Es donde se definen los atributos y métodos que va a tener un objeto representado mediante la misma

**b. Objeto:** Es una instancia de una clase declarada como variable para poder acceder a los atributos y comportamientos que fueron definidos en ella para determinado objeto

**c. Clase Abstracta:** Es una clase donde se define qué métodos y atributos puede tener una clase hija, no es necesario dejar todos los métodos sin implementar, es decir también además de indicar qué métodos deben implementar las clases hijas también puede implementar sus propios métodos.

**d. Interfaz:** Es una clase donde todos sus métodos están sin implementar, únicamente definiendo así la estructura de las clases hijas.

**e. Herencia:** Es añadir a una clase hija todos los métodos y atributos que fueron creados en una clase padre de manera que no es necesario escribir de nuevo los mismos métodos o atributos en diferentes clases que pueden compartir el mismo

**f. Polimorfismo:** Es cuando en las clases hijas el mismo método heredado de la clase padre es sobre escrito para que se comporte de forma diferente a como lo hacen las demás, varias clases pueden compartir un mismo comportamiento pere ejecutarlo de forma diferente

**2.**

Para realizar una aplicación CRUD en codeigniter se debe descargar el framework de codeIgniter y agregarlo a nuestro proyecto como base para partir a crear nuestros propios componentes de acuerdo al modelo MVC, antes de crearlos se debe configurar las credenciales de acceso a la base de datos en el archivo application/config/database.php, una URl de acceso a la aplicación en application/config/config.php y librerías que se usarán globalmente en application/config/autoload.php.

Primero se debe crear un **modelo** desde el que se accede a la base de datos (realizar consultas ya sean obtener, crear, actualizar o eliminar datos) se puede crear un método para cada acción del modelo CRUD en ella.

Segundo se debe crear una **vista** en la que mediante un formulario podremos mostrar y obtener los datos que se usarán para ejecutar los procesos del modelo CRUD, por ejemplo un formulario con una tabla para mostrar los datos de las tablas SQL ó uno con campos de textos para digitar los datos que se van a agregar o actualizar en la tabla. Desde la **vista** también se podrían llamar los métodos definidos en el **controlador** para comunicar la **vista** con el **modelo**

Por último, se debe crear un **controlador** que recibirá las peticiones de la vista, procesa los datos y lo enviará al **modelo** para que este ejecute las consultas SQL, el **modelo** puede retornar valores de acuerdo al resultado de la ejecución de las consultas, por ejemplo si una actualización falla puede responder al controlador para que este redirija a una página de error o a una página mostrando los datos actualizados. Al igual que en el modelo, se crea un método por cada acción que deba ejecutarse (en este caso consultar, agregar, actualizar y eliminar datos de una tabla de la base de datos)