El constructor canónico no puede ser genérico ni tener cláusulas throws, incluso para excepciones no verificadas (unchecked).

Los métodos accessor generados automáticamente no deben lanzar excepciones.

Se pueden definir otros métodos dentro del Record, como en cualquier otra clase.

Es un error de compilación declarar un componente de Record con nombres reservados como clone, finalize, getClass, hashCode, notify, notifyAll, toString, o wait, ya que estos nombres corresponden a métodos de la clase Object.

A diferencia de las clases regulares, donde el compilador no proporciona un constructor por defecto si se declara cualquier otro constructor, los Records siempre reciben un constructor canónico automáticamente, incluso si se definen otros constructores.

Constructores y Métodos:

**Diferencias con Clases** 

Regulares:

Características Adicionales y Limitaciones

Conceptos Clave sobre los Records en Java Definición y Funcionamiento de los Records

especial en Java diseñada para representar un simple conjunto de valores (un agregado simple de datos). Los Records facilitan la creación de clases que solo contienen datos, eliminando gran parte del código repetitivo asociado con los POJOs (Plain Qué es un Record: Old Java Objects). La declaración de un Record define automáticamente campos de instancia que son private y final. También crea métodos **accessor** (getters) públicos para estos campos. Los componentes del Record son los campos definidos en el encabezado del Record y los métodos accessor correspondientes. Ejemplo de declaración de un Record:public record Student(int Componentes del Record: id, String name) {} Este Record es equivalente a una clase que extiende Record e incluye campos private final, un constructor canónico que inicializa los campos, y métodos accessor que devuelven los valores de estos campos. Constructor Canónico: Es el constructor que toma como parámetros los componentes del Record y los inicializa. Si no se declara explícitamente, el compilador lo genera automáticamente. Constructor Compacto: Es una forma abreviada de definir un constructor canónico que permite agregar lógica adicional (como validaciones) sin necesidad de especificar una **Constructores y Métodos** lista de parámetros.public Canónicos: record Student(int id, String name) { **Constructores No Canónicos:** Se pueden definir otros constructores, pero deben llamar explícitamente al constructor canónico o a otro constructor en la primera línea. Los Records pueden definirse a nivel superior (dentro de un paquete), como clases miembro anidadas dentro de otra clase o interfaz, o como clases locales dentro de un método. Las clases anidadas de Record son implícitamente static, aunque es permitido especificar static redundante. Reglas Básicas de los Los Records son implícitamente Records: final y no pueden ser abstract, sealed o non-sealed. No pueden extender otras clases, ya que todos los Records extienden implícitamente java.lang.Record. Un Record no puede contener declaraciones de campos de instancia (solo se permiten campos estáticos). No puede tener inicializadores Restricciones en el Cuerpo de instancia (solo inicializadores del Record: estáticos). No puede tener métodos abstract o native. Los Records se deserializan utilizando el constructor canónico, lo que los diferencia de otros objetos serializables en Serialización de los Records: Java.

Un **Record** es una clase