

ENTERPRISE JAVA DEVELOPER

# JAVA ORIENTADO A OBJETOS

## COLECCIONES

Eric Gustavo Coronel Castillo  
[gcoronelc.blogspot.com](http://gcoronelc.blogspot.com)





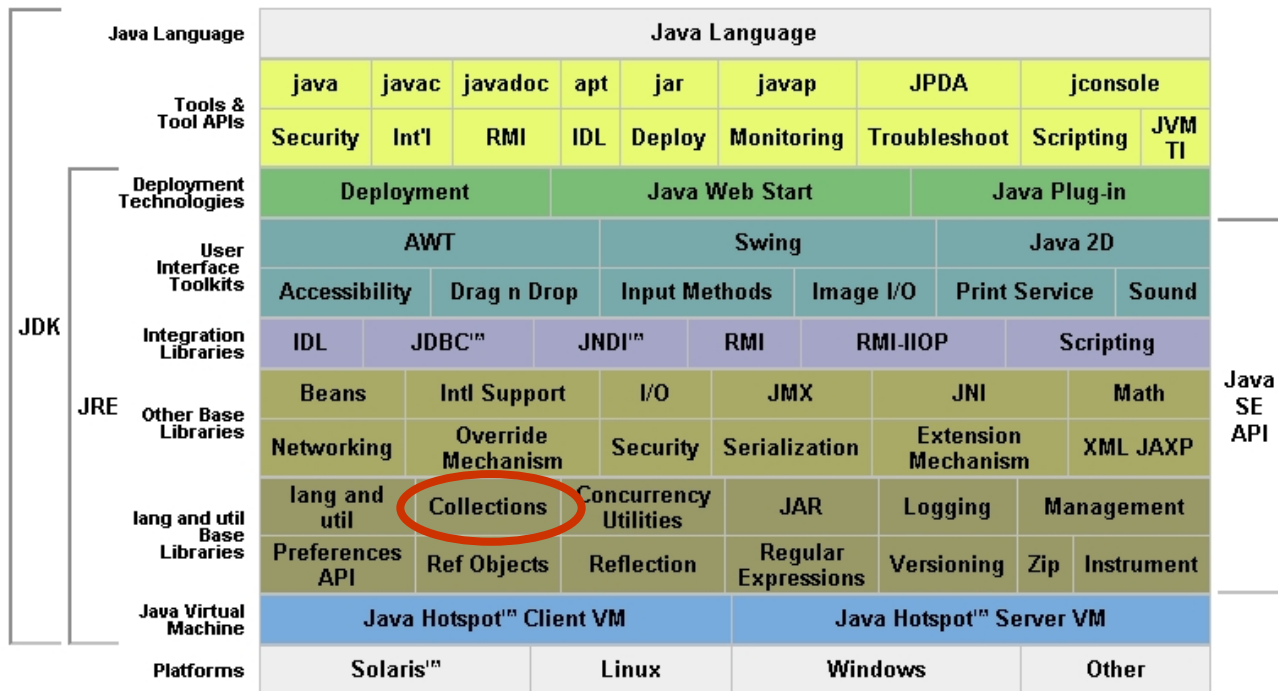
## Temas

- Objetivo
- Introducción
- Interfaces del Core
- Implementación del Core
- Proyecto Ejemplo



# Objetivo

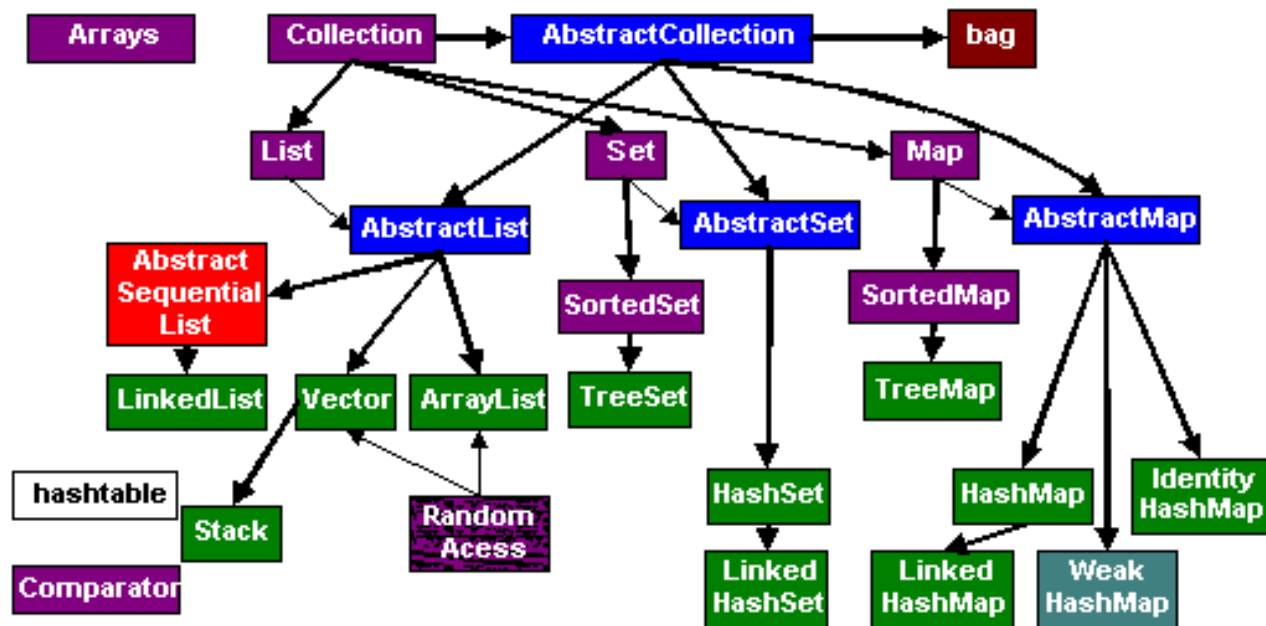
Trabajar colección de datos mediante JCF.





# Introducción

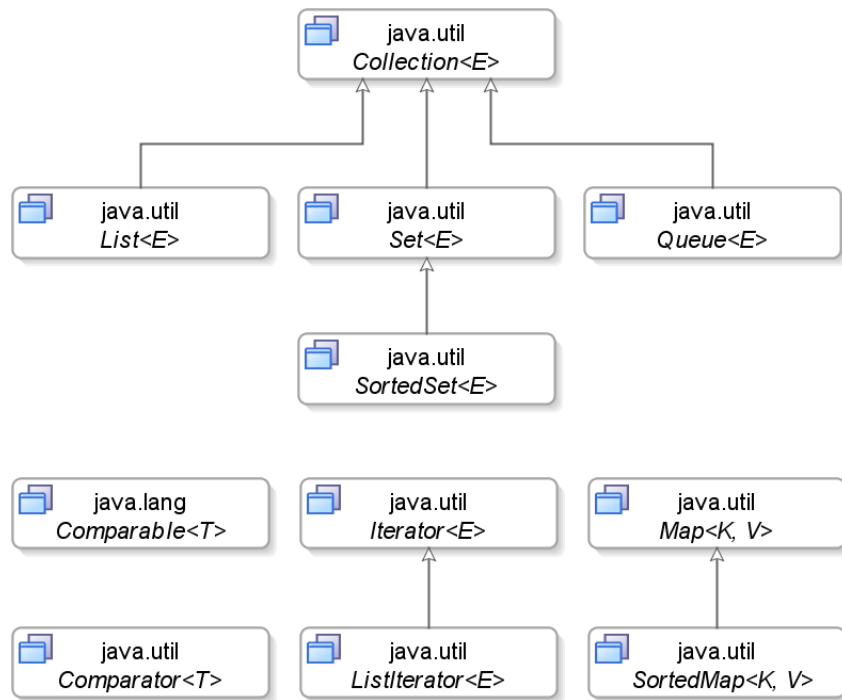
- Un marco de trabajo es un conjunto de interfaces y clases proporcionados para resolver un problema determinado.
- La intención es utilizar las clases proporcionadas, extenderlas o implementar las interfaces.





# Interfaces del Core

Proporcionan métodos para todas las operaciones comunes, y las implementaciones concretas especifican la decisión de las operaciones no permitidas.





# Interfaces del Core

## ■ Interfaces del Core

INTERFAZ	DESCRIPCIÓN
<b>Collection</b>	Representa un grupo de objetos sin implementaciones directas, agrupa la funcionalidad general que todas las colecciones ofrecen.
<b>Set</b>	Colección que no puede tener objetos duplicados.
<b>SortedSet</b>	<a href="#">Set</a> que mantiene los elementos ordenados
<b>List</b>	Colección ordenada que puede tener objetos duplicados
<b>Map</b>	Colección que enlaza claves y valores; no puede tener claves duplicadas y cada clave debe tener al menos un valor.
<b>SortedMap</b>	<a href="#">Map</a> que mantiene las claves ordenadas.
<b>Queue</b>	Colección que maneja la prioridad para procesar los elementos



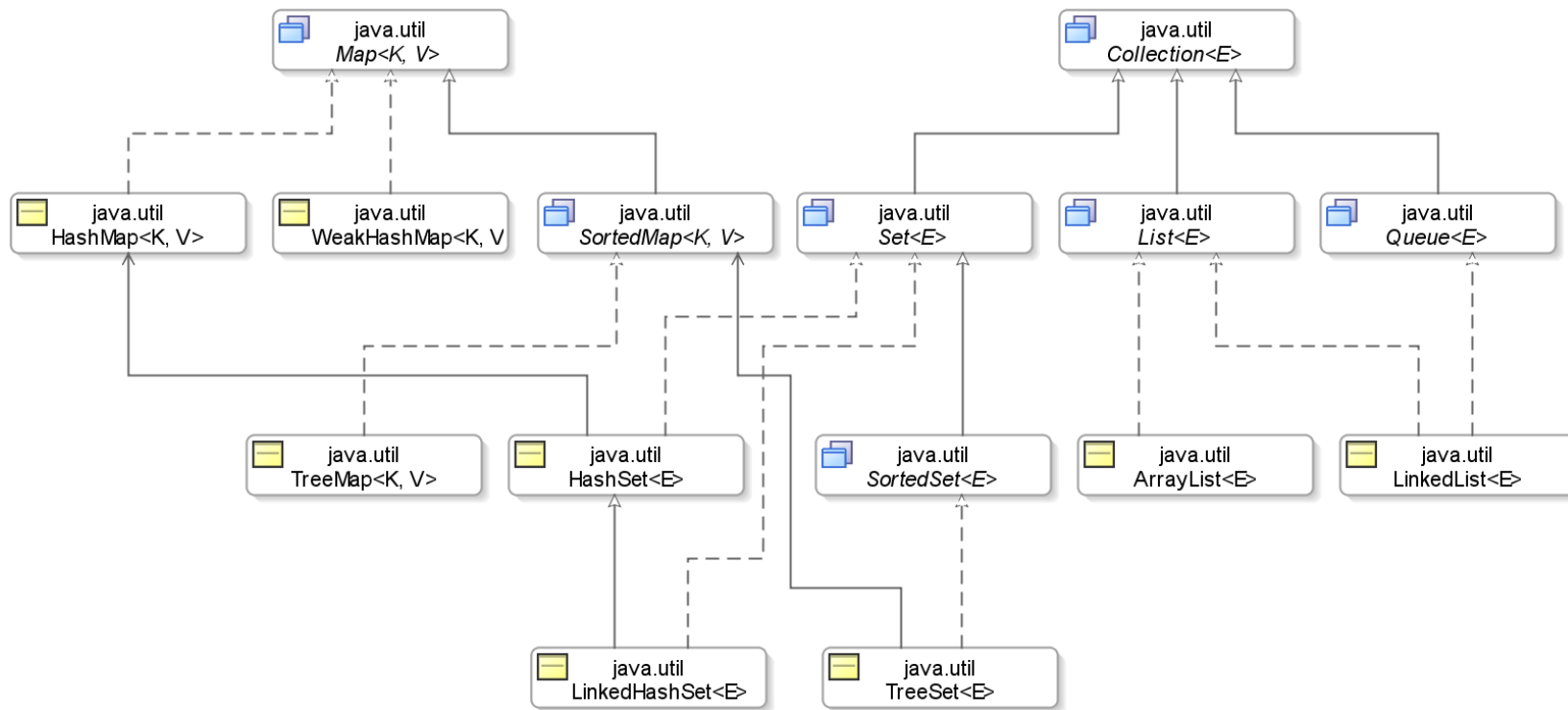
# Interfaces del Core

## ■ Interfaces de Soporte

INTERFAZ	DESCRIPCIÓN
<b>Iterator</b>	Sustituye a la interfaz <a href="#">Enumeration</a> . Dispone de métodos para recorrer una colección y para borrar elementos.
<b>ListIterator</b>	Deriva de <a href="#">Iterator</a> y permite recorrer <a href="#">lists</a> en ambos sentidos.
<b>Comparable</b>	Declara el método <a href="#">compareTo()</a> que permite ordenar las distintas colecciones según un orden natural (String, Date, Integer, Double, ...).
<b>Comparator</b>	Declara el método <a href="#">compare()</a> y se utiliza en lugar de <a href="#">Comparable</a> cuando se desea ordenar objetos no estándar o sustituir a dicha interfaz.



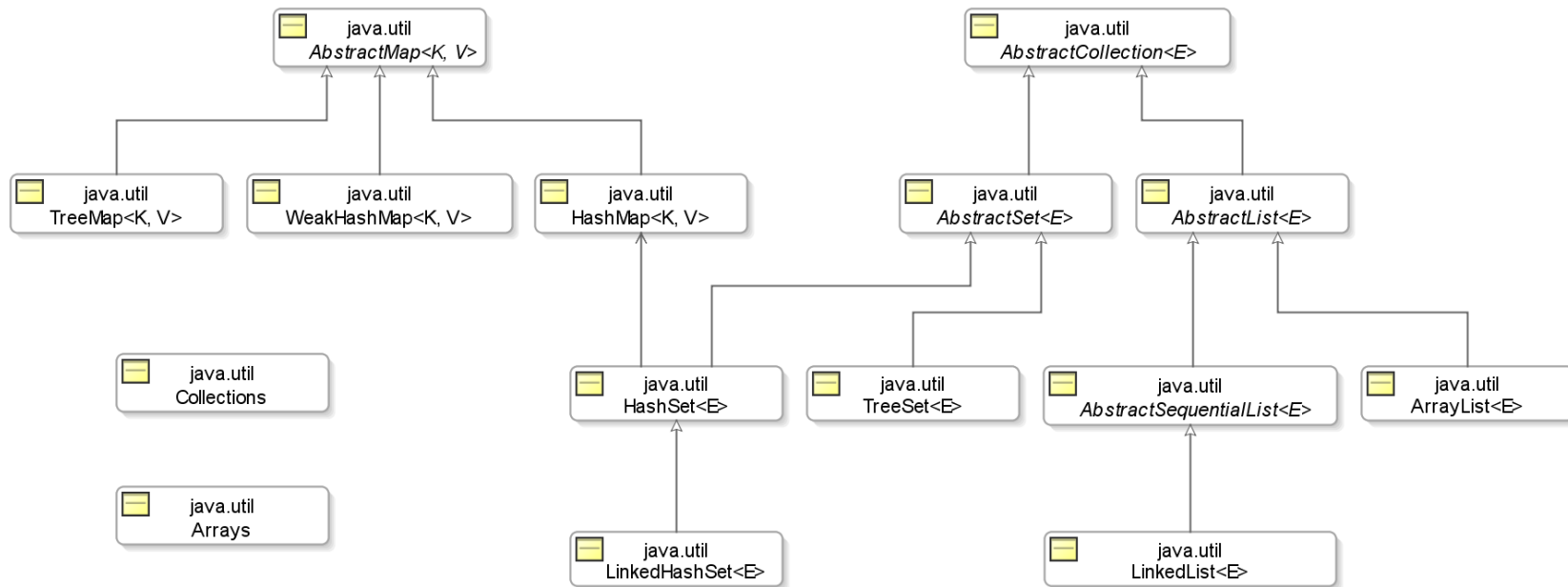
# Implementación del Core







# Implementación del Core





# Implementación del Core

## ■ Clases de Propósito General

CLASE	DESCRIPCIÓN
<b>HashSet</b>	Implementación de la interfaz <a href="#">Set</a> mediante una hash table.
<b>TreeSet</b>	Implementación de la interfaz <a href="#">SortedSet</a> mediante un árbol binario ordenado.
<b>ArrayList</b>	Implementación de la interfaz <a href="#">List</a> mediante un array.
<b>LinkedList</b>	Implementación de la interfaz <a href="#">List</a> mediante una lista vinculada.
<b>HashMap</b>	Implementa la interfaz <a href="#">Map</a> mediante una hash table.
<b>WeakHashMap</b>	Implementa la interfaz <a href="#">Map</a> de modo que la memoria de los pares clave/valor pueda ser liberada cuando las claves no tengan referencia desde el exterior de la <a href="#">WeakHashMap</a> .
<b>TreeMap</b>	Implementa la interfaz <a href="#">SortedMap</a> mediante un árbol binario.



# Manejo de Listas

---

- Creación de una lista

```
List<Producto> lista = new ArrayList<>();
```

- Agregar Elementos

```
lista . add ( new Producto( "Televisor", 890.0, 256 ) );  
lista . add ( new Producto( "Lavadora", 1500.0, 435 ) );
```

- Recorrer los elementos de una lista

```
for ( Producto prod : Lista ) {  
    . . .  
}
```



# Manejo de Datos tipo Clave / Valor

---

- Creación de una lista

```
Map<String,Producto> lista = new HashMap<>();
```

- Agregar Elementos

```
lista . put("Televisor", new Producto("Televidor", 879.99, 450));  
lista . put("Lavadora", new Producto("Lavadora", 1435.67, 236));
```

- Recorrer los elementos de una lista

```
for( String clave: lista.keySet() ){  
    Producto prod = lista.get(clave);  
    . . .  
}
```



## PROYECTO EJEMPLO

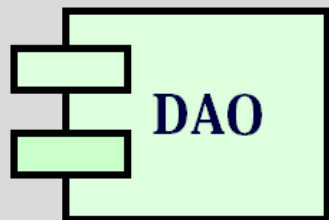
---

- La tienda WilsonStore necesita una aplicación para registrar sus ventas del día.
- Al finalizar el día, necesita los siguientes reportes:
  - Listado de todas las ventas.
  - Resumen de ventas por artículo.
  - Resumen de ventas por categoría.

## CODIGO FUENTE

## EUREKA-CS-ORACLE-JDBC

### APLICACIÓN JAVA



JDBC

A light blue rounded rectangular box with a dark blue border, representing the JDBC driver component.

CONEXIÓN

A large, light gray double-headed arrow with a black outline, indicating a bidirectional connection between the application and the database.

Esquema

EUREKA

ORACLE XE 11g

Dirección de descarga: <https://goo.gl/TDgc5R>

ENTERPRISE JAVA DEVELOPER

# JAVA ORIENTADO A OBJETOS

**Gracias**

Eric Gustavo Coronel Castillo  
[gcoronelc.blogspot.com](http://gcoronelc.blogspot.com)

