GUSTAVO CORONEL
DESARROLLA SOFTWARE

SQL SERVER - IMPLEMENTACIÓN

LECCIÓN 06

CREANDO RESUMENES DE DATOS

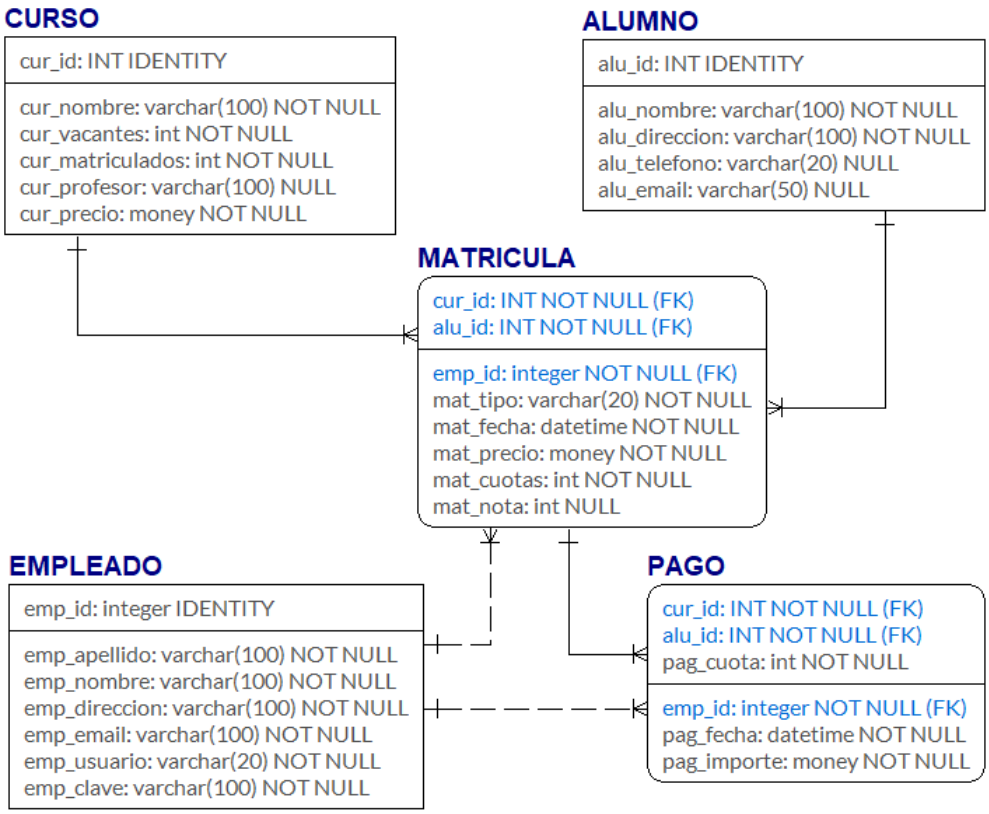
Eric Gustavo Coronel Castillo
youtube.com/DesarrollaSoftware
gcoronel@uni.edu.pe

INDICE

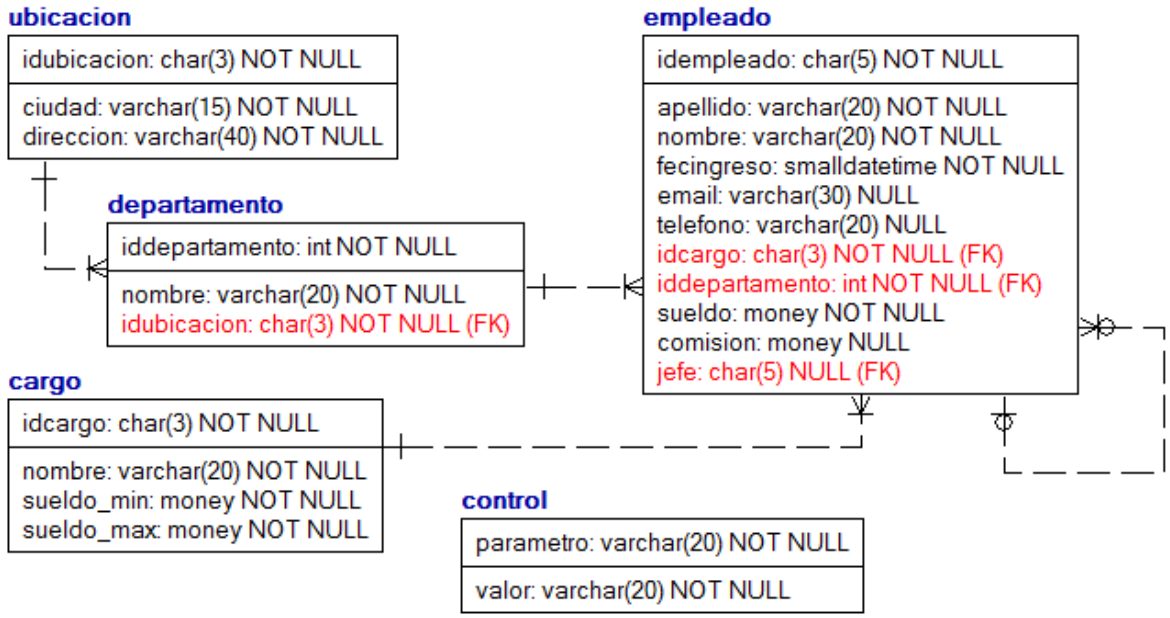
MODELO DE DATOS

Para el desarrollo de la presente lección se utilizan diferentes bases de datos que se ilustran a continuación.

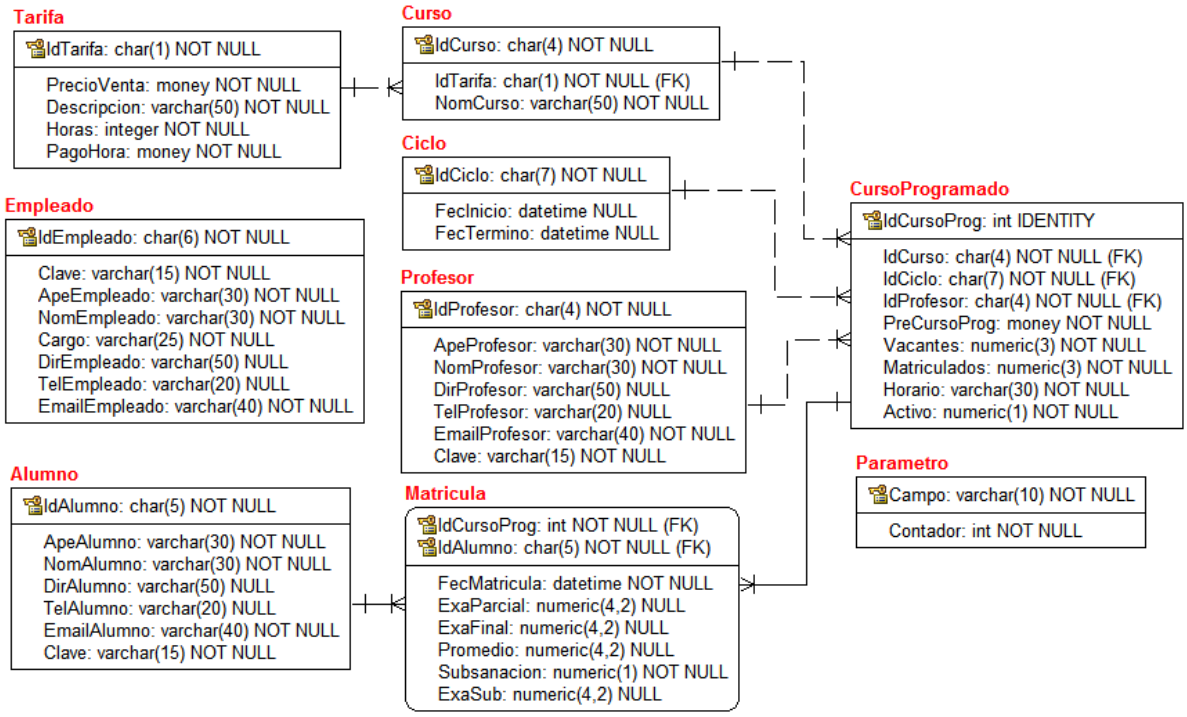
Modelo Educativo Básico (EDUCA2)



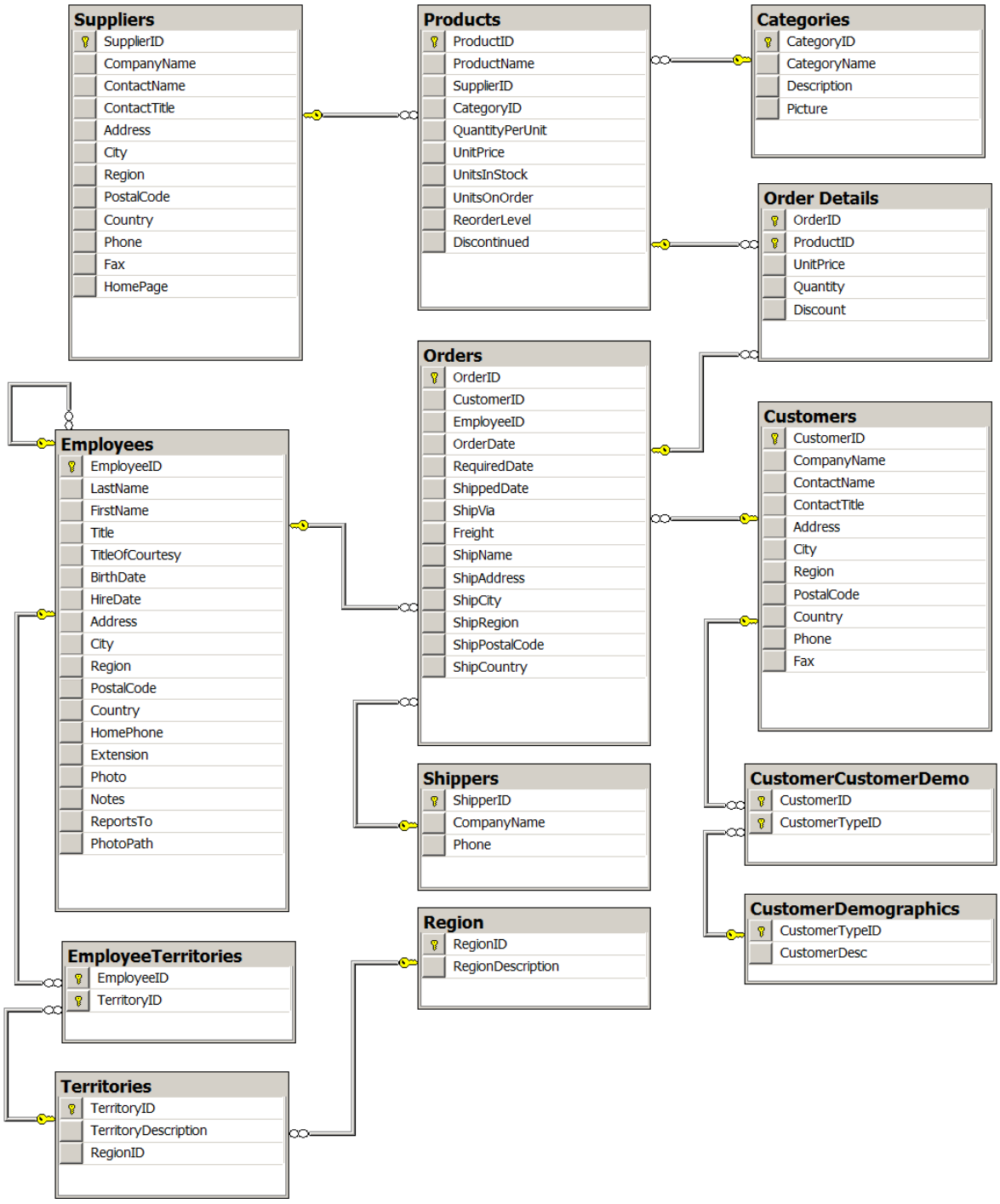
Modelo de recursos humanos (RH)



Modelo académico EDUTEC



Modelo de Gestión de Pedidos (NorthWind)



FUNCIONES DE AGREGACIÓN

A continuación, se tiene una tabla con las funciones agregadas de la sentencia SELECT en SQL Server, su descripción y ejemplos de uso:

Función	Descripción	Ejemplo de uso
SUM()	Calcula la suma de una columna numérica.	SELECT SUM(UnitPrice) FROM Products;
AVG()	Calcula el promedio de una columna numérica.	SELECT AVG(UnitPrice) FROM Products;
COUNT()	Cuenta la cantidad de filas o valores no nulos.	SELECT COUNT(*) FROM Orders;
MIN()	Obtiene el valor mínimo de una columna.	SELECT MIN(UnitPrice) FROM Products;
MAX()	Obtiene el valor máximo de una columna.	SELECT MAX(UnitPrice) FROM Products;
COUNT(DISTINCT)	Cuenta los valores distintos en una columna.	SELECT COUNT(DISTINCT CustomerID) FROM Orders;
STDEV()	Calcula la desviación estándar de una columna numérica.	SELECT STDEV(UnitPrice) FROM Products;
VAR()	Calcula la varianza de una columna numérica.	SELECT VAR(UnitPrice) FROM Products;

Notas:

- Estas funciones se usan comúnmente con GROUP BY para agrupar datos.
- COUNT(*) cuenta todas las filas, mientras que COUNT(columna) solo cuenta los valores no nulos.
- SUM(), AVG(), MIN() y MAX() solo aplican a columnas numéricas.

1. Desarrolle una sentencia SELECT para calcular el importe de la planilla del departamento de ventas. Debe incluir el sueldo y la comisión. Base de datos RH.

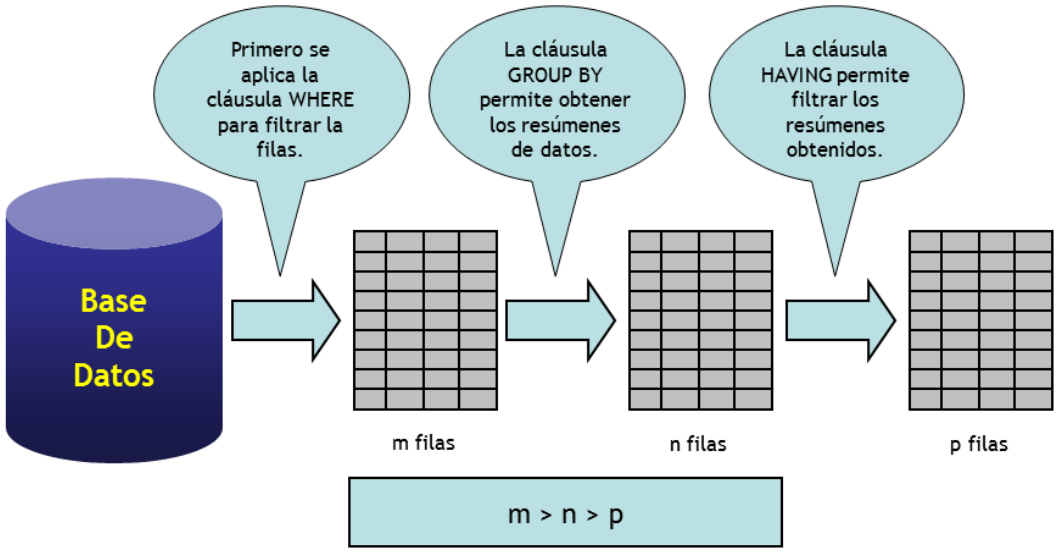
```
SELECT SUM(sueldo + ISNULL(comision,0.0)) PLANILLA
FROM RH..empleado
WHERE iddepartamento=103;
GO
```

2. Desarrolle una sentencia SELECT para encontrar el mayor y menor sueldo en el departamento de ventas. Base de datos RH.

```
SELECT
  MAX(sueldo) [SUELDO MAYOR],
  MIN(sueldo) [SUELDO MENOR]
FROM RH..empleado
WHERE iddepartamento=103;
GO
```

3. Desarrolle una sentencia SELECT para encontrar el salario promedio en la empresa. Base de datos RH.
4. Desarrollar una sentencia SELECT que permita obtener el importe de lo recaudado hasta el momento por los cursos vendidos. Base de datos EDUCA2.
5. Desarrollar una sentencia SELECT que permita consultar el importe de lo que se tiene comprometido (cobrado y no cobrado) por los cursos vendidos hasta el momento. Base de datos EDUCA2.
6. Desarrollar una sentencia SELECT que permita consultar el precio promedio de todos los productos en la tabla Products. Base de datos NorthWind.
7. Desarrollar una sentencia SELECT que permita consultar el número total de registros en la tabla Orders, es decir, el total de pedidos registrados. Base de datos NorthWind.
8. Desarrollar una sentencia SELECT que permita consultar la sumatoria del stock disponible de todos los productos en la tabla Products. Base de datos NorthWind.
9. Desarrollar una sentencia SELECT que permita consultar el precio más alto y más bajo de los productos en la tabla Products. Base de datos NorthWind.

GROUP BY y HAVING



Sintaxis

```
SELECT columna1, columna2, FUN_AGREGADA(columna3)
FROM tabla
WHERE condición
GROUP BY columna1, columna2
HAVING condición_agregada
ORDER BY columna1;
```

Explicación:

- **SELECT columna1, columna2, FUN_AGREGADA (columna3)**: Se seleccionan columnas y una o más funciones agregadas (SUM(), AVG(), COUNT(), etc.).
- **FROM tabla**: Se indica la tabla origen.
- **WHERE condición**: Para especificar el filtro que se debe aplicar antes de la agrupación.
- **GROUP BY columna1, columna2**: Agrupa los resultados según las columnas especificadas.
- **HAVING condición_agregada**: Filtra grupos después de aplicar las funciones agregadas.
- **ORDER BY columna1**: Ordena los resultados finales.

10. Desarrolle una sentencia SELECT para encontrar el salario promedio por departamento. Base de datos RH.

```
SELECT iddepartamento, AVG(sueldo) [SALARIO PROMEDIO]
FROM RH..empleado
GROUP BY iddepartamento;
GO
```

11. Desarrolle una sentencia SELECT para encontrar la cantidad de trabajadores por departamento. Base de datos RH.

```
SELECT iddepartamento, COUNT(1) [CANT TRABAJADORES]
FROM RH..empleado
GROUP BY iddepartamento;
GO
```

12. Desarrolle una sentencia SELECT para encontrar los departamentos que tienen más de 5 trabajadores. Base de datos RH.

```
SELECT iddepartamento, COUNT(1) [CANT TRABAJADORES]
FROM RH..empleado
GROUP BY iddepartamento
HAVING COUNT(1) > 5;
GO
```

13. Desarrolle una sentencia SELECT para encontrar el importe recaudado por curso. Base de datos EDUCA.
14. Desarrolle una sentencia SELECT para encontrar la cantidad de matriculados por curso. Base de datos EDUCA.
15. Desarrollar una sentencia SELECT para encontrar los profesores que tienen más de 2 cursos programados en el ciclo actual. Base de datos EDUTEC.
16. Desarrollar una sentencia SELECT para encontrar la cantidad de veces que se ha programado cada curso en cada ciclo en el presente año. Solo se debe mostrar los cursos que se han programado mas de 2 veces. Base de datos EDUTEC.
17. Desarrollar una sentencia SELECT para encontrar la cantidad de alumnos matriculados por curso en el ciclo actual. Base de datos EDUTEC.
18. Desarrollar una sentencia SELECT para encontrar la cantidad de productos por categoría. Base de datos NorthWind.

19. Desarrollar una sentencia SELECT para encontrar el precio promedio por categoría. Base de datos NorthWind.
20. Desarrollar una sentencia SELECT para encontrar las categorías cuyo precio promedio es mayor a $25. Base de datos NorthWind.
21. Desarrollar una sentencia SELECT para encontrar el stock por categoría. Base de datos NorthWind.
22. Desarrollar una sentencia SELECT para encontrar la cantidad de unidades vendidas por articulo y el importe que representa por artículo. Base de datos NorthWind.
23. Desarrollar una sentencia SELECT para consultar los empleados que han gestionado más de 100 pedidos. Base de datos NorthWind.
24. Desarrollar una sentencia SELECT que permita averiguar los productos que han sido solicitados más de 40 veces. Base de datos NorthWind.

OPERADORES: ROLLUP, CUBE Y GROUPING SETS

OPERADOR	DESCRIPCIÓN	JERARQUÍA EN AGRUPACIÓN	RESULTADOS ADICIONALES
ROLLUP	Genera agregaciones jerárquicas desde un nivel detallado hasta un nivel superior	Sí, sigue un orden jerárquico	Agrega totales parciales y un total general
CUBE	Genera todas las combinaciones posibles de agregación	No sigue una jerarquía fija	Genera subtotales para todas las combinaciones posibles de las columnas agrupadas
GROUPING SETS	Permite definir manualmente los grupos de agregación deseados	Personalizado según la definición	Solo incluye los conjuntos especificados, sin agregar subtotales adicionales

Ejemplo de ROLLUP

Totales jerárquicos: Año → Mes → Total General

```
SELECT
  YEAR(OrderDate) AS Año,
  MONTH(OrderDate) AS Mes,
  COUNT(OrderID) AS TotalPedidos
FROM Orders
GROUP BY ROLLUP(YEAR(OrderDate), MONTH(OrderDate))
ORDER BY Año, Mes;
GO
```

Resultados:

- Muestra la cantidad de pedidos por mes y por año.
- Agrega totales por año.
- Agrega un total general de todos los pedidos.

Ejemplo de CUBE

Todas las combinaciones posibles: Año ↔ Mes

```
SELECT
  YEAR(OrderDate) AS Año,
  MONTH(OrderDate) AS Mes,
  COUNT(OrderID) AS TotalPedidos
FROM Orders
GROUP BY CUBE(YEAR(OrderDate), MONTH(OrderDate))
ORDER BY Año, Mes;
GO
```

Resultados:

- Genera subtotales por año, mes, año y mes combinados.
- Agrega un total general de todos los pedidos.
- No respeta jerarquía, genera combinaciones que ROLLUP no haría.

Ejemplo de GROUPING SETS

Personalización de agregaciones

```
SELECT
  YEAR(OrderDate) AS Año,
  MONTH(OrderDate) AS Mes,
  COUNT(OrderID) AS TotalPedidos
FROM Orders
GROUP BY GROUPING SETS (
  (YEAR(OrderDate), MONTH(OrderDate)),
  (YEAR(OrderDate)),
  ()
)
ORDER BY Año, Mes;
GO
```

Resultados:

- Personaliza los totales, generando solo los grupos deseados.
- Evita generar combinaciones innecesarias como CUBE.

25. Desarrolle una sentencia SELECT para encontrar los ingresos por mes y los ingresos totales. Base de datos EDUCA.

    Aplique: GROUP BY ROLLUP (A, B, . . .)

26. Desarrolle una sentencia SELECT para encontrar el importe de la planilla por cargo en cada departamento, el total por cargo, el total por departamento y el total general. Base de datos RH

    Aplique: GROUP BY ROLLUP (A, B, . . .)

27. Desarrolle una sentencia SELECT para encontrar el importe de la planilla por cargo y departamento, encontrando resúmenes por todas las combinaciones posibles de estos datos. Base de datos RH.

    Aplique: GROUP BY CUBE (A, B, . . .)

28. Desarrolle una sentencia SELECT para encontrar el importe de la planilla por cada cargo, y el importe de la planilla por cada departamento. Base de datos RH.

    Aplique: GROUP BY GROUPING SETS (A, B, . . .)

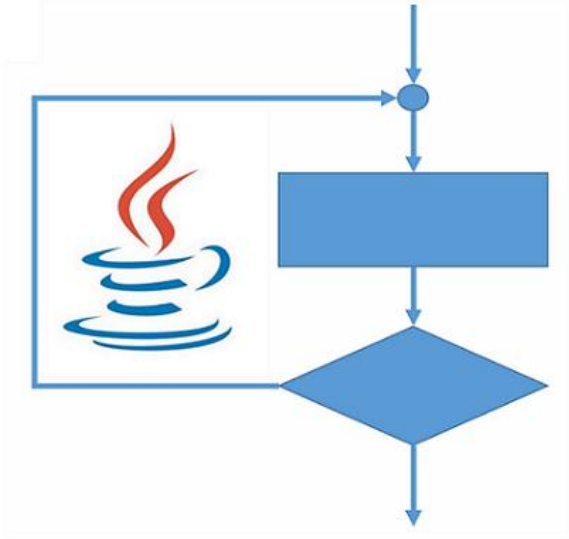
CURSOS VIRTUALES

Acceso a los Cursos Virtuales

En esta URL tienes los accesos a los cursos virtuales:

http://gcoronelc.github.io

Fundamentos de Programación con Java



Tener bases sólidas de programación muchas veces no es fácil, creo que es principalmente por que en algún momento de tu aprendizaje mezclas la entrada de datos con el proceso de los mismos, o mezclas el proceso con la salida o reporte, esto te lleva a utilizar malas prácticas de programación que luego te serán muy difíciles de superar.

En este curso aprenderás las mejores prácticas de programación para que te inicies con éxito en este competitivo mundo del desarrollo de software.

URL del Curso: https://n9.cl/gcoronelc-java-fund

Avance del curso: https://n9.cl/gcoronelc-fp-avance

Cupones de descuento: http://gcoronelc.github.io

Java Orientado a Objetos

CURSO PROFESIONAL DE JAVA ORIENTADO A OBJETOS

Eric Gustavo Coronel Castillo
www.desarrollasoftware.com
INSTRUCTOR

En este curso aprenderás a crear software aplicando la Orientación a Objetos, la programación en capas, el uso de patrones de software y Swing.

Cada tema está desarrollado con ejemplos que demuestran los conceptos teóricos y finalizan con un proyecto aplicativo.

URL del Curso: https://bit.ly/2B3ixUW

Avance del curso: https://bit.ly/2RYGXIt

Cupones de descuento: http://gcoronelc.github.io

Programación con Java JDBC

En este curso aprenderás a programas bases de datos Oracle con JDBC utilizando los objetos Statement, PreparedStatement, CallableStatement y a programar transacciones correctamente teniendo en cuenta su rendimiento y concurrencia.

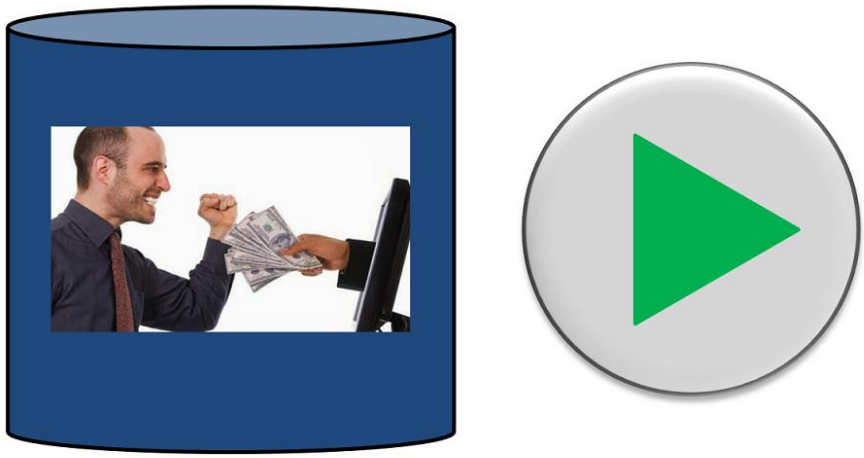
Al final del curso se integra todo lo desarrollado en una aplicación de escritorio.

URL del Curso: https://bit.ly/31apy0O

Avance del curso: https://bit.ly/2vatZOT

Cupones de descuento: http://gcoronelc.github.io

Programación con Oracle PL/SQL



En este curso aprenderás a programas las bases de datos ORACLE con PL/SQL, de esta manera estarás aprovechando las ventas que brinda este motor de base de datos y mejoraras el rendimiento de tus consultas, transacciones y la concurrencia.

Los procedimientos almacenados que desarrolles con PL/SQL se pueden ejecutarlo de Java, C#, PHP y otros lenguajes de programación.

URL del Curso: https://bit.ly/2YZjfxT

Avance del curso: https://bit.ly/3bciqYb

Cupones de descuento: http://gcoronelc.github.io