---========================================================================

---Punto 1

**SELECT** p.ProductID , p.NameProduct , p.Price

**FROM** Products p

**Where** p.Price > 1000

---========================================================================

---Punto 2

**SELECT** c.CustomerID , c.NameCustomer , ih.total

**FROM** Customers c

**INNER** **JOIN** InvoiceHead ih **ON** c.CustomerID = ih.CustomerID

---========================================================================

---Punto 3

**SELECT** ih.numberID, c.NameCustomer , ih.DateInvoice , p.NameProduct , id.Quantity , p.Price

**FROM** InvoiceHead ih

**INNER** **JOIN** Customers c **ON** ih.CustomerID = c.CustomerID

**INNER** **JOIN** InvoiceDetail id **ON** ih.numberID = id.NumberID

**INNER** **JOIN** Products p **ON** id.ProductID = p.ProductID

---========================================================================

---Punto 4

**SELECT** c.CustomerID, c.NameCustomer

**FROM** Customers c

**WHERE** **NOT** **EXISTS** (**SELECT** ih.CustomerID **FROM** InvoiceHead ih

**WHERE** c.CustomerID = ih.CustomerID)

---========================================================================

---Punto 5

**SELECT** **TOP** 3 id.ProductID , p.NameProduct, **COUNT**(id.ProductID) **as** Total

**FROM** InvoiceDetail id

**INNER** **JOIN** Products p **ON** id.ProductID = p.ProductID

**GROUP** **BY** id.ProductID, p.NameProduct

**ORDER** **by** **COUNT**(id.ProductID) **DESC**

---========================================================================

---Punto 1

**SELECT** p.PRODUCTO , p.NOBRE\_PRODUCTO , p.VALOR

**FROM** PRODUCTOS p

**Where** p.VALOR > 1000

---========================================================================

---Punto 2

**SELECT** c.CLIENTE , c.NOMBRE\_CLIENTE , ih.TOTAL

**FROM** CLIENTES c

**INNER** **JOIN** CABEZA\_FACTURA ih **ON** c.CLIENTE = ih.CLIENTE

---========================================================================

---Punto 3

**SELECT** ih.NUMERO, c.NOMBRE\_CLIENTE , ih.FECHA , p.NOMBRE\_PRODUCTO , id.CANTIDAD , p.VALOR

**FROM** InvoiceHead ih

**INNER** **JOIN** CLIENTES c **ON** ih.CLIENTE = c.CLIENTE

**INNER** **JOIN** DETALLE\_FACTURA id **ON** ih.NUMERO = id.NUMERO

**INNER** **JOIN** PRODUCTOS p **ON** id.PRODUCTO = p.PRODUCTO

---========================================================================

---Punto 4

**SELECT** c.CLIENTES, c.NOMBRE\_CLIENTES

**FROM** CLIENTES c

**WHERE** **NOT** **EXISTS** (**SELECT** ih.CLIENTES **FROM** CABEZA\_FACTURA ih

**WHERE** c.CLIENTES = ih.CLIENTES

---========================================================================

---Punto 5

**SELECT** **TOP** 3 id.PRODUCTO, p.NOMBRE\_PRODUCTO, **COUNT**(id. PRODUCTO) **as** Total

**FROM** DETALLE\_FACTURA id

**INNER** **JOIN** Products p **ON** id.PRODUCTO = p.PRODUCTO

**GROUP** **BY** id.PRODUCTO, p.NOMBRE\_PRODUCTO

Conceptos de lenguajes de desarrollo

**Punto a**

El diseño estructurado de sistemas se ocupa de la identificación, selección y organización de los módulos y sus relaciones.

**Punto b.**

Se le llama así a cualquier lenguaje de programación que implemente los conceptos definidos por la programación orientada a objetos.

Donde este es un modelo de programación informática que organiza el diseño de software en torno a datos u objetos, en lugar de funciones y lógica.

**Punto c.**

**Clase:** definiciones de las propiedades y comportamiento de un tipo de objeto concreto.

Diagrama

Descripción generada automáticamente

Ejemplo en Java

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Objeto:** Instancia de una clase. Entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos), los mismos que consecuentemente reaccionan a eventos.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Método**: algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un "mensaje". Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.

Texto

Descripción generada automáticamente

**Evento**: es un suceso en el sistema (tal como una interacción del usuario con la máquina, o un mensaje enviado por un objeto). El sistema maneja el evento enviando el mensaje adecuado al objeto pertinente. También se puede definir como evento la reacción que puede desencadenar un objeto; es decir, la acción que genera.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Punto d**

Diferencias clave

* C es un lenguaje de programación de computadoras orientado a procedimientos y funciones, mientras que Java es un lenguaje de programación de computadoras orientado a objetos.
* C es un lenguaje compilado, mientras que Java es un lenguaje interpretado.
* Java se deriva de C ya que la mayor parte de su sintaxis se deriva de C y C ++. Java depende de C mientras que C es independiente.
* Aunque las características de sintaxis de Java se derivan de C, sigue siendo un lenguaje de más alto nivel.
* C usa el enfoque de arriba hacia abajo (enfoque nítido y suave) mientras que JAVA usa el enfoque de abajo hacia arriba (en las rocas).
* La sobrecarga de métodos también es la diferencia entre estos dos lenguajes. Java admite la sobrecarga de métodos que C no admite en absoluto.
* Los preprocesadores son compatibles con C, no con Java en absoluto.
* Java puede manejar los errores y cualquier interrupción usando técnicas de manejo de excepciones, mientras que C no es bueno para manejar errores.
* Pascal se puede tipificar a los lenguajes de programación de alto nivel según la manera que trabajan los programas que los utilizan y la filosofía de su creación
* El Pascal esta comprendidos en este grupo los nuevos lenguajes que intentan aproximar el diseño y la construcción de programas informáticos al lenguaje utilizado por las personas.

**Punto e**

* Eclipse
* NetBeans
* JeatBrains(PHPStorm, PyCharm, IntelliJ IDEA, etc)
* Visual Studio y Visual Studio Code
* ATOM

**Punto 2**