|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **INSTITUTO DE EDUCACIÓN SUPERIOR CIBERTEC**  **DIRECCIÓN ACADÉMICA**  **CARRERA PROFESIONALES** | | | |  |
|  | | | |  |
| **CURSO** | **:** | **Base de Datos Avanzado II (2398)** | NOTA | |
| **PROFESOR** | **:** | **JESUS CANALES GUANDO** |
| **CICLO** | **:** | **Cuarto** |
| **SECCIÓN** | **:** | **T4CJ** |
| **GRUPO** | **:** | **01** |
| **FECHA** | **:** | **Miércoles 23 de abril del 2025** |
| **DURACIÓN** | **:** | **90 minutos** |
|  |  |  |  | |
| **ALUMNO (A)** | **:** | Ronald Jaime Duran | | |

**EXAMEN DE LABORATORIO 2 (T2)**

**Consideraciones generales:**

* Considerar el orden, la limpieza y la claridad de las respuestas.
* NO está permitido el uso o consulta de cuadernos, separatas, libros o cualquier material de la asignatura durante el desarrollo de la evaluación. El desarrollo es estrictamente personal.
* Todas las **respuestas** deberán aparecer dentro del espacio de cada pregunta, dichas **respuestas** deben evidenciar el script y sus resultados.
* Puede utilizar Mongo Shell, Compass o Studio 3T.
* Toda copia parcial o total será amonestado con la nota 99, **sin** posibilidad a recuperación.
* El desarrollo del presente examen no está permitido enviar por correo, sólo es permitido por el Blackboard.

|  |
| --- |
| **LOGRO DE LA EVALUACIÓN**  Tenga presente que el logro, es que usted**,** aplique operadores en la implementación de consultas de documentos, así como el uso de condicionales y ordenamientos. |

**CONSOLIDADO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pregunta** | **Puntaje** | | **Llenar solo en caso de Recalificación justificada** | |
| **Máximo** | **Obtenido** | **Sustento** | **Puntaje** |
| **1** | **04** |  |  |  |
| **2** | **04** |  |  |  |
| **3** | **06** |  |  |  |
| **4** | **06** |  |  |  |
|  | **20** |  |  |  |
| **Nota Recalificada** | | | |  |

**NOTA: Para evaluar su desarrollo debe pegar su script y captura de pantalla mostrando el resultado de ejecución como evidencia por cada pregunta. En toda captura de su desarrollo debe figurar sus nombres completos, por ejemplo “Por Juan Pérez”. Asegurarse que toda respuesta sea legible.**

*Implemente la base de datos NoSQL en* ***MongoDB*** *llamada* ***“UrbanStyle\_Store”*** *con las colecciones:* ***clientes, productos, vendedores y******sucursales*** *con sus documentos.* ***Script*** *proporcionado por el docente.*

Implementar consultas de documentos de las colecciones:

**PREGUNTA 1.-**

**Obtenga de acuerdo al caso las consultas solicitadas de la colección “clients”, aplicando:**

**Operadores de comparación:** $eq, $gt, $gte, $in, $lt, $lte, $ne, $nin u otros:

|  |  |  |
| --- | --- | --- |
| Id | Descripción | Puntaje |
| 1 | Muestre los **clientes** que no residan en Breña ni en La Molina. Ordene en descendente por apellidos. Sólo debe mostrar los campos: nombres, apellidos, email e historial de compras. |  |
| Script: | use("UrbanStyle\_store");    db.clients.find(      {          direccion:          {              $not: { $regex: /(la molina|breña)/i }          }      },  {      \_id:0,      nombre: 1,      apellidos: 1,      email:1,      telefono:1,      direccion:1,      preferencias:1,      historialCompras: 1  }  ).sort({nombre: -1}); | 1 |
| Captura: |  | 1 |
| 2 | Muestre los **clientes** que tienen como parte de sus preferencias ropa de niños. Considere visualizar sólo los campos: nombres, apellidos, fecha de registro y preferencias. Ordene por fecha de registros de manera ascendente. |  |
| Script: | use("UrbanStyle\_store");  db.clients.find(      {          preferencias:          {              $in: ["Niño" ]          }      },      {          \_id: 0,          nombre: 1,          apellidos: 1,          fechaRegistro: 1,          preferencias:1      }  ).sort({ "fechaRegistro": 1 }) | 1 |
| Captura: |  | 1 |

**PREGUNTA 2.-**

**Recupere documentos solicitados de la colección “productos”, según sea el caso utilizando:**

**Operadores lógicos:** $and, $or, $not, $nor, u otros:

|  |  |  |
| --- | --- | --- |
| Id | Descripción | Puntaje |
| 1 | Listar de los productos \_id, nombre, marca, talla y precio de las prendas, filtre cuya sucursal sea “Sucursal Norte” y el precio sea mayor igual a 100. |  |
| Script: | use("UrbanStyle\_store");    db.products.find(      {          $and:          [              { sucursal: {$eq: "Sucursal Norte" }} ,              { precio: {$gte: 45}}          ]      },      {          \_id: 1,          nombre: 1,          marca: 1,          talla: 1,          precio: 1      }    ); | 1 |
| Captura: | No había productos con precio >= a 100 y que sean de la sucursal norte  Por lo que ajuste el precio a >= 45 para que se pueda ver el resultado | 1 |
| 2 | Mostrar los productos cuyo precio sea mayor a 55 o color “azul”, no considerar de la temporada “Invierno” |  |
| Script: | use("UrbanStyle\_store");  db.products.find(    {      $and: [        {          "especificaciones.temporada": {$ne: "Invierno"}        },        {          $or:          [            { precio: {$gt: 55} },            { "especificaciones.color" : { $eq: "Azul"} },          ]        }      ]    }  );  use("UrbanStyle\_store");    db.products.find(); | 1 |
| Captura: |  | 1 |

**PREGUNTA 3.-**

**Recupere documentos según sea el caso utilizando los métodos:** sort(), limit(), skip(), $and así como

**Operadores de formación (array):** $all, $elemMatch, $size, $slice u otros métodos:

|  |  |  |
| --- | --- | --- |
| Id | Descripción | Puntaje |
| 1 | Utilizando la colección vendedores mostrar nombres, ventas mensuales y sede en la que labora, la lista debe mostrar después del tercer documento, filtre por sede que sea la “Sucursal Sur”. Ordene por cualquier campo. |  |
| Script: | use("UrbanStyle\_store");    db.sellers.find(    {          sucursal: {$eq: "Sucursal Sur"}    },    {      \_id: 0,      nombre: 1,      "metas.ventasMensuales": 1,      sucursal: 1    }    ).skip(3).sort({ nombre: 1}); | 1 |
| Captura: | Profesor la consulta sin skip() obtiene 3 documentos    pero al utilizar skip omite esos 3 documentos, dando así vacío como respuesta | 1 |
| 2 | Listar sólo los nombres, apellidos y fecha de ingreso (sin “\_id”), ordenada en descendente por apellidos, recuperar los 4 primeros documentos. Filtre sólo los vendedores cuya cantidad de clientes nuevos que atienden sean menores e iguales que 20. |  |
| Script: | db.sellers.find(      {          "metas.clientesNuevos": { $lte: 20}      },      {          \_id: 0,          nombre: 1,          apellidos: 1,          fechaIngreso:1,      }    ).sort({apellidos: -1}).limit(4); | 1 |
| Captura: |  | 1 |
| 3 | Mostrar todos aquellos vendedores en los que dentro de su campo “ventas mensuales” sea mayor e igual que 10000 y menor que 11000. Ordene en descendente por nombre. |  |
| Script: | use("UrbanStyle\_store");  db.sellers.find({      $and: [      {          "metas.ventasMensuales": { $gte: 10000}      },      {          "metas.ventasMensuales": { $lt: 11000}      }      ]  },  {      \_id: 0,      nombre: 1,      apellidos: 1,      "metas.ventasMensuales": 1    }  ).sort({ nombre: -1 }); | 1 |
| Captura: | **En los documentos no hubieron Arrays, por lo que no fue necesario utilizar consultas de arrays** | 1 |

**PREGUNTA 4.-**

**Se tiene el siguiente comprobante como documento comercial, colección “factura” (no debe crear ninguna BD ni colección ni documentos, SÓLO UNA CONSULTA):**

A paper with a signature

Description automatically generated

**Requerimiento**: Usted como responsable del análisis de datos semiestructurados en mongoDB para un proyecto de TI debe tomar en cuenta la distribución de la información de dicha boleta para luego **crear un reporte** donde considere:

* Visualizar al menos 5 campos.
* Dos filtros.
* Un ordenamiento.

|  |  |  |
| --- | --- | --- |
| Id | Descripción | Puntaje |
| Script: | */\*se lista aquellos productos en el que*  *el precio unitario sea >= 50*  *el total de factura sea  < 100.50*  *ordenado en ascendente por total\_factura*  *\*\*/*  db.productos.find(      {          $and: [              { precio\_unitario: {$gte: 50 } },              { total\_factura: {$lt: 100.50} }          ]      },      {          \_id: 1,          descripcion:1,          precio\_unitario: 1,          importe: 1,          total\_factura:1,          cant\_productos:1      }  ).sort({total\_factura: -1}) | 6 |

**Rúbrica de evaluación**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rúbrica de la Pregunta 1 (Condicionales con operadores lógicos y ordenamiento)** | | | | |
| **Puntos** | **Excelente** | **Bueno** | **Regular** | **Deficiente** |
| 4 | Implementa las consultas correctamente según requerimiento. Demuestra evidencia. | Implementa las consultas según requerimiento. Existe alguna inconsistencia. Demuestra evidencia. | Implementa las consultas según requerimiento. Posee dos a tres errores. | No desarrolla requerimiento alguno o todo presenta errores. |
|  | **4 puntos** | **2 puntos** | **1 punto** | **0 puntos** |
| **Rúbrica de la Pregunta 2 (Condicionales con operadores and y or)** | | | | |
| **Puntos** | **Excelente** | **Bueno** | **Regular** | **Deficiente** |
| 4 | Implementa las consultas correctamente según requerimiento. Demuestra evidencia. | Implementa las consultas según requerimiento. Existe alguna inconsistencia. Demuestra evidencia. | Implementa las consultas según requerimiento. Posee dos a tres errores. | No desarrolla requerimiento alguno o todo presenta errores. |
|  | **4 puntos** | **2 puntos** | **1 punto** | **0 puntos** |
| **Rúbrica de la Pregunta 3 (Condicionales con array o vectores)** | | | | |
| **Puntos** | **Excelente** | **Bueno** | **Regular** | **Deficiente** |
| 6 | Implementa las consultas correctamente según requerimiento. Demuestra evidencia. | Implementa las consultas según requerimiento. Existe alguna inconsistencia. Demuestra evidencia. | Implementa las consultas según requerimiento. Posee dos a tres errores. | No desarrolla requerimiento alguno o todo presenta errores. |
|  | **6 puntos** | **4 puntos** | **2 puntos** | **0 puntos** |
| **Rúbrica de la Pregunta 4 (Análisis de datos semiestructurados y consulta)** | | | | |
| **Puntos** | **Excelente** | **Bueno** | **Regular** | **Deficiente** |
| 6 | Implementa la consulta correctamente según requerimiento. | Implementa la consulta según requerimiento. Posee un error. | Implementa la consulta según requerimiento. Posee dos a tres errores. | No desarrolla requerimiento alguno o todo presenta errores. |
|  | **6 puntos** | **4 puntos** | **2 puntos** | **0 puntos** |