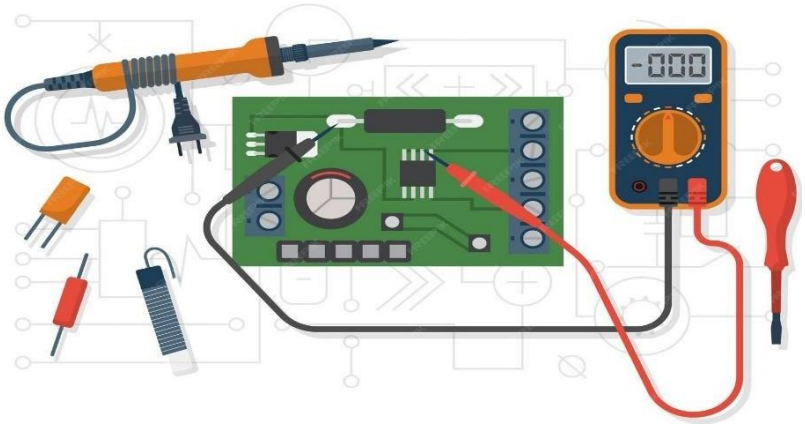
<table>
<tr><td>EVALUACIÓN</td><td>Obligatorio</td><td>GRUPOS</td><td>todos</td><td>FECHA</td><td>Marzo 2025</td></tr>
<tr><td>MATERIA</td><td colspan="5">Bases de Datos 2</td></tr>
<tr><td>CARRERA</td><td colspan="5">Analista en Tecnologías de Información / Analista Programador</td></tr>
<tr><td>CONDICIONES</td><td colspan="5">
- Puntaje máximo: 40 puntos<br>
- Puntaje mínimo: 1 punto<br>
- Fecha de entrega: 23/06/2025 hasta las 21:00 horas en gestion.ort.edu.uy (max. 40Mb en formato zip, rar o pdf)<br><br>
Uso de material de apoyo y/o consulta<br><br>
Inteligencia Artificial Generativa<br>
- Seguir las pautas de los docentes: Se deben seguir las instrucciones específicas de los docentes sobre cómo utilizar la IA en cada curso.<br>
- Citar correctamente las fuentes y usos de IA: Siempre que se utilice una herramienta de IA para generar contenido, se debe citar adecuadamente la fuente y la forma en que se utilizó.<br>
- Verificar el contenido generado por la IA: No todo el contenido generado por la IA es correcto o preciso. Es esencial que los estudiantes verifiquen la información antes de usarla.<br>
- Ser responsables con el uso de la IA: Conocer los riesgos y desafíos, como la creación de “alucinaciones”, los peligros para la privacidad, las cuestiones de propiedad intelectual, los sesgos inherentes y la producción de contenido falso<br>
- En caso de existir dudas sobre la autoría, plagio o uso no atribuido de IAG, el docente tendrá la opción de convocar al equipo de obligatorio a una defensa específica e individual sobre el tema<br><br>
IMPORTANTE:<br><br>
1) Inscribirse<br>
2) Formar grupos de hasta 2 personas del mismo dictado<br>
3) Subir el trabajo a Gestión antes de la hora indicada (ver hoja al final del documento: “RECORDATORIO”)<br><br>
Aquellos de ustedes que presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos, por favor contactarse con el Coordinador o Coordinación adjunta antes de las 20:00hs. del día de la entrega, a través de los mails alamon@ort.edu.uy y rodriguez_mb@ort.edu.uy, o telefónicamente al 29021505 - int 1156, 1138 u 1135.
</td></tr>
</table>

La empresa ***DeviceRepair*** dedicada a reparar dispositivos electrónicos y productos de la industria, posee un modelo resumido de su base de datos (se adjunta el script completo en un anexo).



**Empleado** (***IdEmp***, NomEmp, FchNacEmp, SueldoEmp, TipoEmp)
Los empleados pueden ser de dos tipos, técnicos "T" o controllers "C", si son técnicos participan en la reparación de equipos, si son controllers son los que hacen el control de calidad de las reparaciones (QA).

**Producto** (***IdProd***, Dscprod, StkProd, CostoProd)
Cada producto tiene un identificador autonumérico y siempre se conoce tanto su descripción, el stock y el costo de los mismos.

**Unidad** (***NumSerie,IdProd***, FchFab, FchVto)
Son las unidades de cada producto, están identificadas con un número de serie dependiente del producto, también se conoce la fecha de fabricación y la fecha de vencimiento del producto.

**Repara** (***IdRepara***, NumSerie, IdProd, IdEmp, FchRepara, CostoRepara IdEmpQA)
Las reparaciones de las unidades de cada producto por parte de los empleados técnicos, están identificadas con un autonumérico, se conocen los datos, se conoce además de los datos identificatorios de las unidades, el identificador del técnico que repara, la fecha el costo y el identificador del técnico que realiza el control de calidad.

**Se Pide:**

1. Creación de índices que considere puedan ser útiles para optimizar las consultas (según criterio establecido en el curso).

2. Ingreso de un juego completo de datos de prueba (será más valorada la calidad de los datos que la cantidad).

3. Utilizando SQL implementar las siguientes consultas:

    a. Para todos los productos existentes, mostrar código y descripción, cantidad de reparaciones con control de calidad realizado, cantidad sin control realizado y cantidad de reparaciones cuyo valor fue superior a $100.

    b. Muestra los datos del Empleado con mayor cantidad de reparaciones realizadas.

    c. Muestra datos del producto y costo total de reparaciones por producto, mostrando solo los productos con un costo total superior a $200.

    d. Datos del producto más reparado.

    e. Escribe una consulta que muestre información detallada de los empleados, incluyendo:

        Clasificación del salario en tres niveles (Alto, Medio, Bajo)
        Categoría del empleado según su tipo (Tiempo Completo o Contratado) y nivel salarial (Senior, Junior, Experimentado).
        Cantidad de reparaciones realizadas por cada empleado.
        Ordenar la consulta por salario en orden descendente y, en caso de empate, por el número de reparaciones en orden descendente.

        Elige los rangos de clasificación de acuerdo con tu criterio (justifica).

    f. Muestra el costo total de reparaciones por empleado y un resumen general.

    g. Muestra los datos de los Empleados Técnicos que repararon **todos** los productos.

4. Utilizando T-SQL realizar los siguientes ejercicios:

    a. Crea un procedimiento almacenado llamado *sp_RegistrarReparacion*, que permita registrar una nueva reparación en la tabla Repara, cumpliendo con las siguientes reglas:

    Parámetros de entrada:

    @NumSerie (char(10)): Número de serie de la unidad.
    @IdProd (int): ID del producto asociado.
    @IdEmp (int): ID del empleado que realiza la reparación.
    @CostoRepara (money): Costo de la reparación.

    Validaciones:

    La unidad (@NumSerie, @IdProd) debe existir en la tabla Unidad.
    El empleado (@IdEmp) debe existir en la tabla Empleado.
    Un empleado no puede registrar más de una reparación para la misma unidad en el mismo día.
    El costo de reparación no puede ser negativo.

    Operación:

    Insertar la nueva reparación en la tabla Repara, con el estado 'Iniciado' y la fecha/hora actual.
    Retornar un mensaje indicando el éxito o el motivo del fallo.

    b. Crea una función escalar llamada *fn_CalcularTiempoReparacion*, que reciba el número de serie y el ID del producto y devuelva el tiempo total (en días) que ha estado en reparación.

    Parámetros de entrada:
    @NumSerie (char(10)): Número de serie de la unidad.
    @IdProd (int): ID del producto.

    Contar todos los días únicos en los que la unidad ha estado en reparación, sin importar el número de reparaciones en un mismo día.
    Usar la columna FchRepara de la tabla Repara.
    Si la unidad no tiene reparaciones registradas, debe devolver NULL.

    Salida:

Un int con la cantidad de días distintos en los que se ha reparado la unidad.

5. Escribir los siguientes disparadores (por supuesto: considerando modificaciones múltiples)

   a. Crea un disparador llamado *trg_ControlEstadoReparacion*, que se active cuando se modifique la columna StsRepara en la tabla Repara.

      El trigger debe ejecutarse cuando:

      El estado de reparación (StsRepara) cambie a "Terminado" o "Cancelado". No debe activarse si el estado no cambió.

      Acciones a realizar:

      Registrar el cambio en una tabla llamada HistoricoReparacion, que debe crearse con los siguientes campos:

      ```
      CREATE TABLE HistoricoReparacion (
      IdHist INT IDENTITY PRIMARY KEY,
      IdRepara INT NOT NULL,
      NumSerie CHAR(10) NOT NULL,
      IdProd INT NOT NULL,
      EstadoAnterior VARCHAR(20) NOT NULL,
      EstadoNuevo VARCHAR(20) NOT NULL,
      FchCambio DATETIME DEFAULT GETDATE());
      ```

      Insertar en esta tabla el IdRepara, NumSerie, IdProd, el estado anterior, el estado nuevo y la fecha del cambio.

   b. Crea un disparador llamado *trg_PrevenirEliminacionReparaciones*, que impida la eliminación de registros en la tabla Repara si la reparación tiene el estado "Terminado" o "En testing".

      El trigger debe activarse cuando se intente eliminar un registro en Repara.
      Debe permitir eliminar reparaciones solo si su estado es "Iniciado" o "Cancelado".
      Si el estado es "En testing" o "Terminado", debe bloquear la eliminación y mostrar un mensaje de error.
      Si la eliminación es permitida, debe registrarse en una tabla de auditoría HistoricoEliminacionReparaciones, con los siguientes datos:

```
CREATE TABLE HistoricoEliminacionReparaciones (
  IdHist INT IDENTITY PRIMARY KEY,
  IdRepara INT NOT NULL,
  NumSerie CHAR(10) NOT NULL,
  IdProd INT NOT NULL,
  StsRepara VARCHAR(20) NOT NULL,
  FchEliminacion DATETIME DEFAULT GETDATE()
);
```

6. Crea una vista llamada *vw_ReparacionesActivas*, que muestre información detallada de las reparaciones en curso, es decir, aquellas cuyo estado sea "Iniciado" o "En testing".

   La vista debe incluir la siguiente información:

   IdRepara (ID de la reparación).
   NumSerie (Número de serie de la unidad).
   IdProd (ID del producto).
   DscProd (Descripción del producto, obtenida de la tabla Producto).
   NomEmp (Nombre del empleado que está realizando la reparación, obtenido de Empleado).
   FchRepara (Fecha de reparación).
   StsRepara (Estado de la reparación).

   Filtrar solo las reparaciones activas, es decir, aquellas con StsRepara = 'Iniciado' o StsRepara = 'En testing'.

7. Como se sabe, la empresa de servicio técnico gestiona reparaciones de dispositivos electrónicos y su base de datos en SQL Server (DeviceService) maneja información estructurada sobre empleados, productos y reparaciones en curso.
   Sin embargo, necesitan almacenar un historial detallado de cada reparación, incluyendo:

   a. Eventos y cambios de estado de la reparación (ejemplo: "Se cambió una pieza", "Se realizó prueba funcional").
   b. Notas de los técnicos sobre el diagnóstico y acciones tomadas.
   c. Imágenes o documentos adjuntos relacionados con la reparación (ejemplo: fotos de fallas, informes de pruebas).

   Dado que estos datos son altamente flexibles y varían entre reparaciones, se decide almacenarlos en MongoDB, mientras que la estructura transaccional sigue en SQL Server.
   Cree una estructura en MongoDB para esta realidad, con dicha estructura el usuario debe poder resolver por lo menos los siguientes escenarios:

   a. Dado un IdRepara, queremos ver todos los eventos asociados a esa reparación, incluyendo las notas y fechas de cada acción realizada.
   b. Queremos encontrar todas las reparaciones donde en los eventos haya una **referencia a "placa"**, por ejemplo, para identificar equipos con problemas en la placa base.
   c. Queremos recuperar todas las reparaciones donde se han adjuntado imágenes del proceso de reparación.
   d. Queremos saber **cuántas reparaciones han pasado por una acción específica**, como "Reemplazo de pieza", para evaluar la frecuencia de este tipo de intervención.

**Consideraciones generales:**

1. Los docentes de la materia cumplirán el rol de usuario final del producto a los efectos de evacuar las dudas que puedan surgir a los estudiantes en detalles que no estén incluidos explícitamente en la letra. Independientemente de esto, los alumnos podrán investigar sobre sistemas existentes, así como aportes basados en su propia experiencia o relevamiento con terceros para enriquecer la solución a los problemas planteados siempre que no contradiga lo explicitado en la letra. Cualquier agregado deberá documentarse claramente en la solución y será considerado positivamente en la evaluación. Modificaciones de la letra que puedan surgir durante el curso, serán publicadas en aulas y deberán considerarse en la entrega final.

2. La corrección del obligatorio se hará en base a la estructura entregada junto con la letra del mismo, por lo que los puntos desarrollados deben ser testeados sobre esta estructura. **Soluciones a los puntos del obligatorio que no ejecuten correctamente sobre la estructura proporcionada serán evaluados como incorrectos.**

3. Durante la última semana los docentes **no** contestarán dudas del Obligatorio por ningún medio. Esta consideración intenta evitar que los alumnos dejen la implementación del obligatorio para último momento. Se insta a los estudiantes a desarrollar el obligatorio durante el transcurso del semestre para entregar un trabajo de calidad.

## Anexo 1 - Script de creación de tablas (puede descargar el archivo aparte)

```
CREATE DATABASE DeviceService
GO
USE DeviceService
GO
CREATE TABLE Empleado(IdEmp int identity not null,
                      NomEmp varchar(30) not null,
                       FchNacEmp date not null,
                       SueldoEmp money,
                       TipoEmp char(1) not null,
                       constraint pk_Empleado primary key(IdEmp),
                       constraint ck_TipoEmp check(TipoEmp in ('T','C')))
GO
CREATE TABLE Producto(IdProd int identity not null,
                      DscProd varchar(30) not null,
                       StkProd int,
                       CostoProd money,
                       constraint pk_Producto primary key(IdProd))
GO
CREATE TABLE Unidad(NumSerie character(10) not null,
                    IdProd int not null,
                    FchFab date,
                    FchVto date,
                    constraint pk_Unidad primary key(NumSerie,IdProd),
                    constraint fk_ProdUnidad foreign key(IdProd)
                             references Producto(IdProd))
GO
CREATE TABLE Repara(IdRepara int identity not null,
                    NumSerie character(10) not null,
                                    IdProd int not null,
                                    IdEmp int not null,
                                    FchRepara datetime not null,
                                    CostoRepara money,
                                    StsRepara varchar(20) default 'Iniciado',
                                    IdEmpQA int,
                                    constraint pk_Repara primary key(IdRepara),
                                    constraint uk_Repara
unique(NumSerie,IdProd,IdEmp,FchRepara),
                                    constraint ck_StsRepara check(StsRepara in
('Iniciado','En testing','Terminado','Cancelado')),
                                    constraint fk_UnidadRepara foreign key(NumSerie,IdProd)
references Unidad(NumSerie,IdProd),
                                    constraint fk_EmpRepara foreign key(IdEmp) references
Empleado(IdEmp),
                                    constraint fk_EmpQA foreign key(IdEmpQA) references
Empleado(IdEmp))
GO
```

**RECORDATORIO: IMPORTANTE PARA LA ENTREGA**

- **Obligatorios**

La entrega de los obligatorios será en formato digital online, a excepción de algunas materias que se entregarán en Bedelía y en ese caso recibirá información específica en el dictado de la misma.

Los principales aspectos a destacar sobre la **entrega online de obligatorios** son:

1. Ingresá al sistema de Gestión.
2. En el menú, seleccioná el ítem "Evaluaciones" y la instancia de evaluación correspondiente, que figura bajo el título "Inscripto".
3. Para iniciar la entrega hacé clic en el ícono:
4. Ingresá el número de estudiante de cada uno de los integrantes y hacé clic en "Agregar". El sistema confirmará que los integrantes estén inscriptos al obligatorio y, de ser así, mostrará el nombre y la fotografía de cada uno de ellos. Una vez agregados todos los integrantes, hacé clic en "Crear equipo".

   **Cualquier integrante podrá:**

   - **Modificar la integración del equipo.**
   - **Subir el archivo de la entrega.**

5. Seleccioná el archivo que deseás entregar. Verificá el nombre del archivo que aparecerá en la pantalla y hacé clic en "Subir" para iniciar la entrega. Cada equipo (hasta 2 estudiantes) debe entregar **un único archivo en formato zip o rar** (los documentos de texto deben ser pdf, y deben ir dentro del zip o rar). El archivo a subir debe tener **un tamaño máximo de 40mb** Cuando el archivo quede subido, se mostrará el nombre generado por el sistema (1), el tamaño y la fecha en que fue subido.
6. El sistema enviará un e-mail a todos los integrantes del equipo informando los detalles del archivo entregado y confirmando que la entrega fue realizada correctamente.
7. Podés cerrar la pestaña de entrega y continuar utilizando Gestión o salir del sistema.
8. La **hora tope para subir el archivo será las 21:00** del día fijado para la entrega.
9. La entrega se podrá realizar desde cualquier lugar (ej. hogar del estudiante, laboratorios de la Universidad, etc).
10. Aquellos de ustedes que presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos, por favor contactarse con la Coordinadora o Coordinación adjunta antes de las 20:00hs. del día de la entrega, a través de los mails, alamon@ort.edu.uy y rodriguez_mb@ort.edu.uy, o telefónicamente al 29021505 - int 1156, 1138 u 1135.