Modelos y bases de datos

Seguridad

CEIS

2020-1

Agenda

Seguridad

Control de acceso
- Obligatorio
- Discrecional

Cifrado

Registro de auditoría

Ejemplos vistas

Agenda

Conceptos

Seguridad

Seguridad vs Integridad

Conceptos

Seguridad

La seguridad se refiere a la protección de los datos contra su revelación, su alteración o su destrucción no autorizadas.

Seguridad vs Integridad

- Seguridad significa proteger los datos ante usuarios no autorizados

  Garantizar que los usuarios tengan permiso de hacer las cosas que están tratando de hacer

- Integridad significa proteger los datos de usuarios autorizados

  Asegurar que las cosas que están tratando de hacer sean correctas

Conceptos

Seguridad

La seguridad se refiere a la protección de los datos contra su revelación, su alteración o su destrucción no autorizadas.

Mecanismos

Conceptos

Seguridad

La seguridad se refiere a la protección de los datos contra su revelación, su alteración o su destrucción no autorizadas.

Mecanismos

- Control de acceso

  Definir explícitamente permisos de acciones sobre elementos determinados

- Cifrado

  Guardar o transmitir la información sensible de manera cifrada

- Registro de auditoría

  Guardar las acciones realizadas por los usuarios

Agenda

Control de acceso

Control de acceso

Definir explícitamente permisos de acciones sobre objetos determinados a personas identificadas (ID.Clave)

Mecanismos

Control de acceso

Control de acceso

Definir explícitamente permisos de acciones sobre objetos determinados a personas identificadas (ID.Clave)

Mecanismos

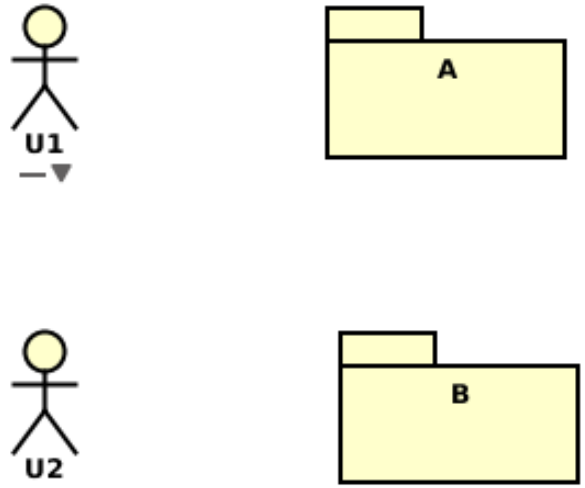
- Discrecional

  Un usuario específico tendrá diferentes niveles de acceso (privilegios) sobre diferentes elementos

- Obligatorio

  Cada objeto de datos está etiquetado con un nivel de clasificación determinado y a cada usuario se le da un nivel de acreditación

Control de acceso



Mecanismos

- Discrecional

  Un usuario específico tendrá diferentes niveles de acceso (privilegios) sobre diferentes elementos

  U1 puede estar autorizado para ver A y no ver B y U2 puede estar autorizado para ver B y no A

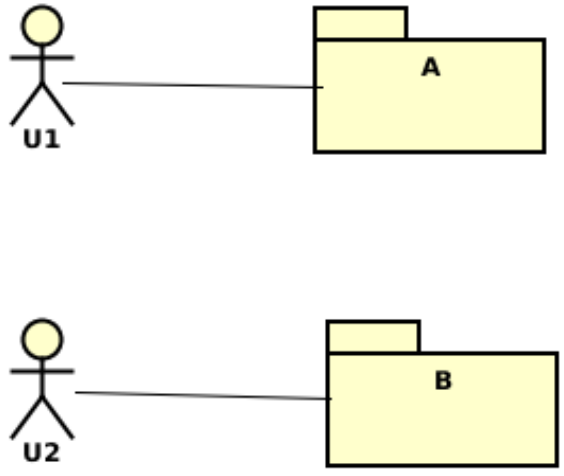
- Obligatorio

  Cada objeto de datos está etiquetado con un nivel de clasificación determinado y a cada usuario se le da un nivel de acreditación

  Si U1 está autorizado para ver A y no ver B entonces nadie podrá ver B y no A

¿POR QUÉ?

Control de acceso



Mecanismos

- Discrecional

  Un usuario específico tendrá diferentes niveles de acceso (privilegios) sobre diferentes elementos

  U1 puede estar autorizado para ver A y no ver B y U2 puede estar autorizado para ver B y no A

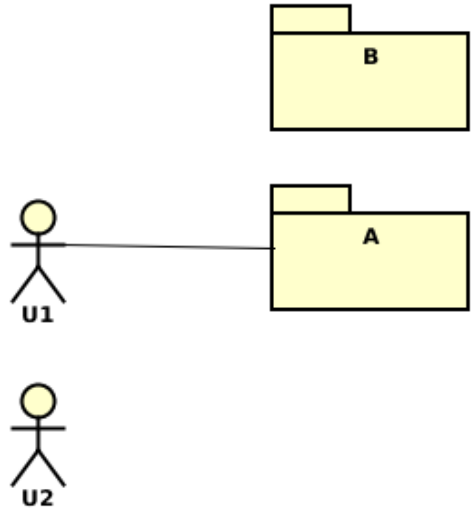
- Obligatorio

  Cada objeto de datos está etiquetado con un nivel de clasificación determinado y a cada usuario se le da un nivel de acreditación

  Si U1 está autorizado para ver A y no ver B entonces nadie podrá ver B y no A

¿POR QUÉ?

Control de acceso



Mecanismos

- Discrecional

  Un usuario específico tendrá diferentes niveles de acceso (privilegios) sobre diferentes elementos

  U1 puede estar autorizado para ver A y no ver B y U2 puede estar autorizado para ver B y no A

- Obligatorio

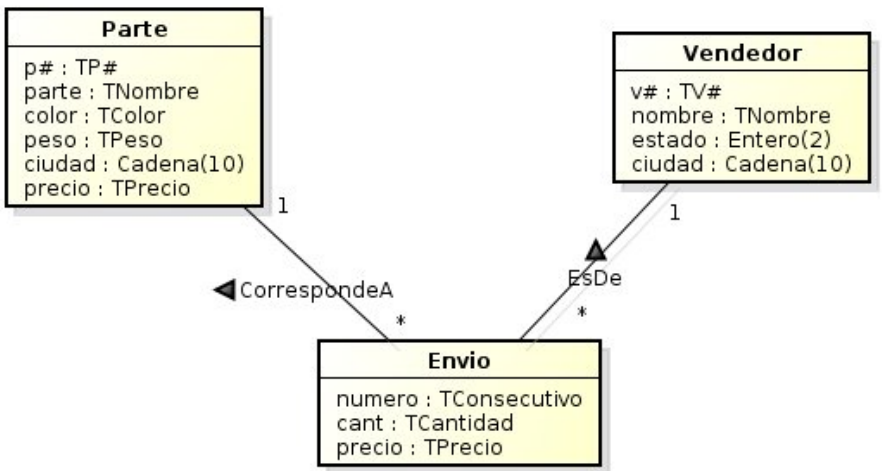
  Cada objeto de datos está etiquetado con un nivel de clasificación determinado y a cada usuario se le da un nivel de acreditación

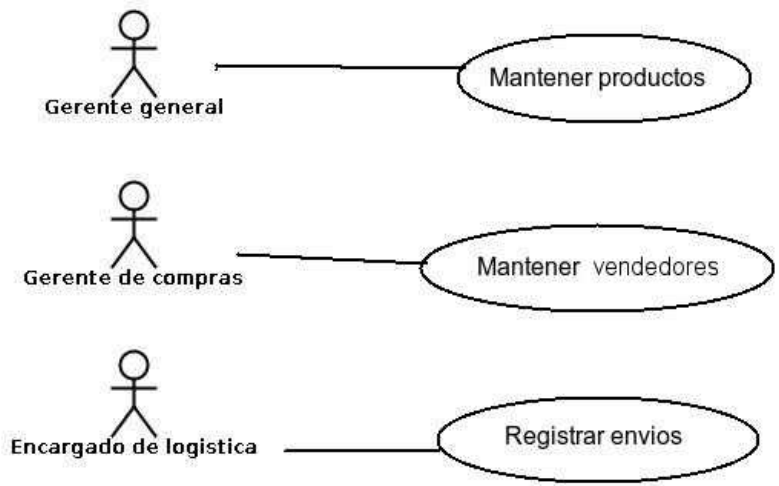
  Si U1 está autorizado para ver A y no ver B entonces nadie podrá ver B y no A

¿POR QUÉ?

Contexto

Envíos

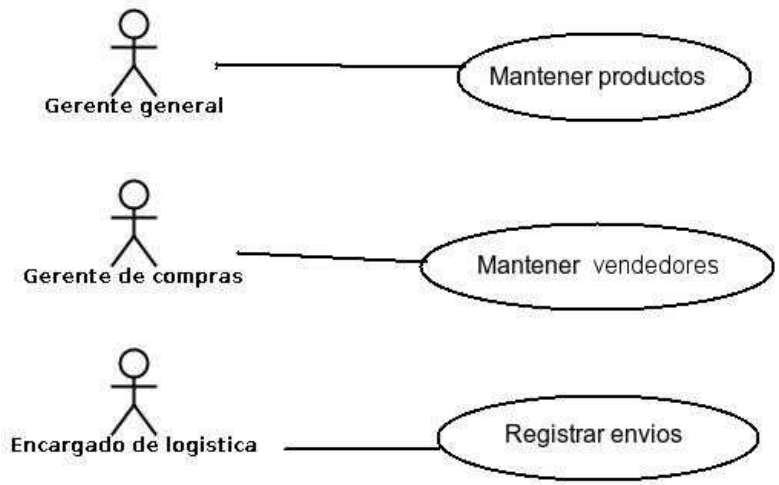




Contexto

Envíos

```
CREATE TABLE VENDEDOR(
  v#          CHAR(2),
  nombre      VARCHAR(20),
  estatus     NUMBER(2),
  ciudad      VARCHAR(10));

CREATE TABLE PARTES(
  p#          CHAR(2),
  parte       VARCHAR(20),
  color       CHAR(10),
  peso        NUMERIC(5,2),
  ciudad      VARCHAR(10));

CREATE TABLE ENVIOS(
  v#          CHAR(2),
  p#          CHAR(2),
  cant        NUMERIC(5));
```



Agenda

Obligatorio

Organización

- Se definen los diferentes niveles
- A cada objeto se le asigna un nivel de clasificación
- A cada usuario se le asigna un nivel de acreditación

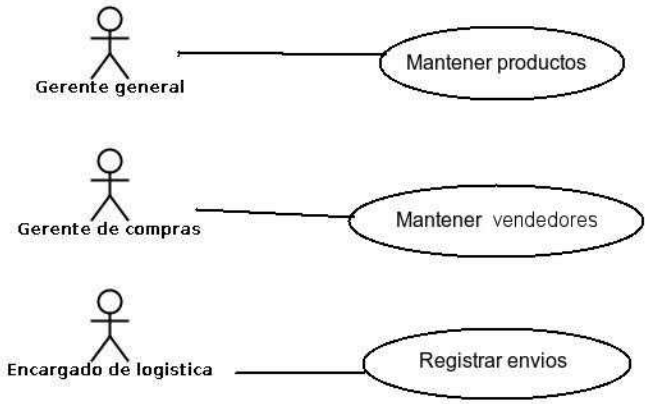
Políticas

1. El usuario i puede recuperar el objeto j sólo si el nivel de acreditación de i es mayor o igual al nivel de seguridad de j
2. El usuario i puede actualizar el objeto j sólo si el nivel de acreditación de i es igual al nivel de clasificación de j

Obligatorio

Envíos

```
CREATE TABLE VENDEDOR(
  v#          CHAR(2),
  nombre      VARCHAR(20),
  estatus     NUMBER(2),
  ciudad      VARCHAR(10));

CREATE TABLE PARTES(
  p#          CHAR(2),
  parte       VARCHAR(20),
  color       CHAR(10),
  peso        NUMERIC(5,2),
  ciudad      VARCHAR(10));

CREATE TABLE ENVIOS(
  v#          CHAR(2),
  p#          CHAR(2),
  cant        NUMERIC(5));
```



Políticas

1. ¿Cuántos niveles?
2. ¿Cuál sería el nivel de cada tabla?
3. ¿Cuál sería el nivel de cada usuario?

   ¿Cómo quedan los permisos?

Agenda

Discrecional

Mecanismos

- Privilegio sobre datos
- Privilegio sobre acciones

Discrecional

Mecanismos

- Privilegio sobre datos
  - Mínimos
  - Tabla
  - Vistas
- Privilegio sobre acciones
  - Subprogramas
  - Paquetes

Discrecional. Privilegios sobre datos.

Dar

```
GRANT privilegios
ON elemento
TO [ usuario | rol | PUBLIC ]
[WITH GRANT OPTION]
```

Quitar

```
REVOKE privilegios
ON elemento
FROM [ usuario | rol | PUBLIC ]
[RESTRICT | CASCADE]
```

Discrecional. Privilegios sobre datos.

Discrecional - Datos

```
GRANT privilegio {, privilegio}
ON [ tabla | vista ]
TO [ usuario | rol | PUBLIC ]
[WITH GRANT OPTION]
```

```
privilegios

INSERT [(columnas)]
DELETE
UPDATE [(columnas)]
SELECT [(columnas)]
ALL
```

Discrecional. Privilegios sobre datos. Mínimos.

Grant

```
CREATE TABLE VENDEDOR(
  v#          CHAR(2),
  nombre      VARCHAR(20),
  estatus     NUMBER(2),
  ciudad      VARCHAR(10));

CREATE TABLE PARTES(
  p#          CHAR(2),
  parte       VARCHAR(20),
  color       CHAR(10),
  peso        NUMERIC(5,2),
  ciudad      VARCHAR(10));

CREATE TABLE ENVIOS(
  v#          CHAR(2),
  p#          CHAR(2),
  cant        NUMERIC(5));
```

```
UPDATE VENDEDORES
SET estatus = estatus + 1
WHERE ((SELECT COUNT(p#) FROM PARTES) =
       (SELECT COUNT(DISTINCT p#) FROM ENVIOS WHERE VENDEDORES.v#=ENVIOS.v#));
```

Privilegios mínimos

- ¿Qué se está haciendo?

Discrecional. Privilegios sobre datos. Mínimos.

Grant

```
CREATE TABLE VENDEDOR(
  v#          CHAR(2),
  nombre      VARCHAR(20),
  estatus     NUMBER(2),
  ciudad      VARCHAR(10));

CREATE TABLE PARTES(
  p#          CHAR(2),
  parte       VARCHAR(20),
  color       CHAR(10),
  peso        NUMERIC(5,2),
  ciudad      VARCHAR(10));

CREATE TABLE ENVIOS(
  v#          CHAR(2),
  p#          CHAR(2),
  cant        NUMERIC(5));
```

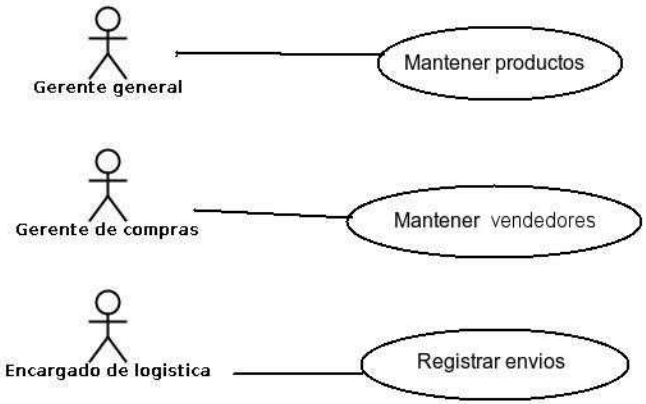
```
UPDATE VENDEDORES
SET estatus = estatus + 1
WHERE ((SELECT COUNT(p#) FROM PARTES) =
       (SELECT COUNT(DISTINCT p#) FROM ENVIOS WHERE VENDEDORES.v#=ENVIOS.v#));
```

Privilegios mínimos

- ¿Qué se está haciendo?
- ¿Cuáles privilegios mínimos debe tener 'ELOGISTICA' para realizar esta actualización?

Discrecional. Privilegios sobre datos. Tablas.

Envíos



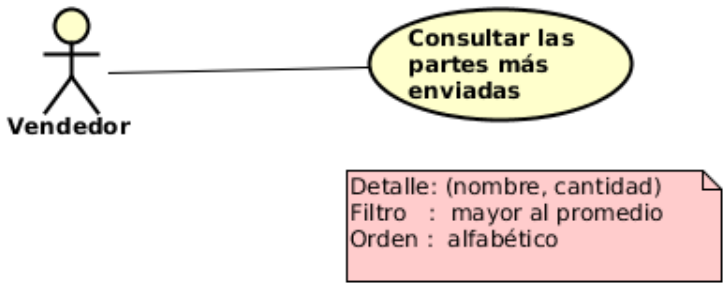

```
CREATE TABLE VENDEDOR(
  v#          CHAR(2),
  nombre      VARCHAR(20),
  estatus     NUMBER(2),
  ciudad      VARCHAR(10));

CREATE TABLE PARTES(
  p#          CHAR(2),
  parte       VARCHAR(20),
  color       CHAR(10),
  peso        NUMERIC(5,2),
  ciudad      VARCHAR(10));

CREATE TABLE ENVIOS(
  v#          CHAR(2),
  p#          CHAR(2),
  cant        NUMERIC(5));
```

Privilegios generales

- ¿Qué permisos daríamos sobre las tablas?

Discrecional. Privilegios sobre datos. Vistas.

Envíos




```
CREATE TABLE VENDEDOR(
  v#          CHAR(2),
  nombre      VARCHAR(20),
  estatus     NUMBER(2),
  ciudad      VARCHAR(10));

CREATE TABLE PARTES(
  p#          CHAR(2),
  parte       VARCHAR(20),
  color       CHAR(10),
  peso        NUMERIC(5,2),
  ciudad      VARCHAR(10));

CREATE TABLE ENVIOS(
  v#          CHAR(2),
  p#          CHAR(2),
  cant        NUMERIC(5));
```

Privilegios generales

- ¿Qué vista definiríamos?
- ¿Qué permisos daríamos sobre esta vista?

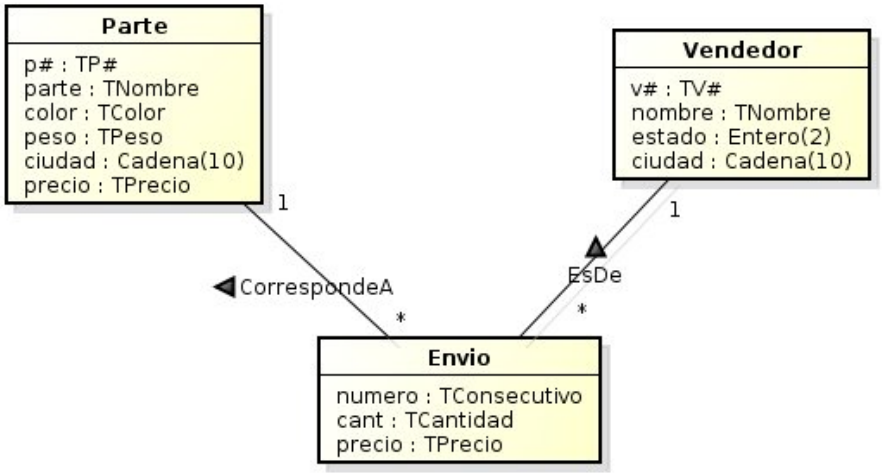
Discrecional. Privilegios de ejecución.

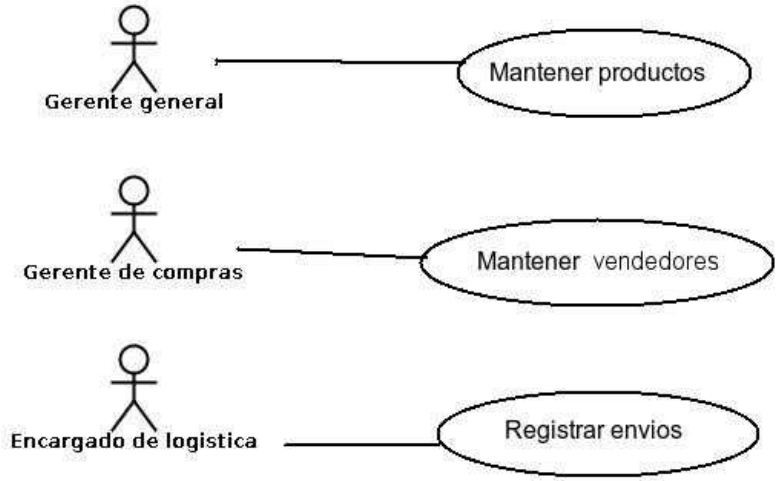
Sobre acciones

```
GRANT EXECUTE
ON [ subprograma | paquete ]
TO [ usuario | rol | PUBLIC ]
[WITH GRANT OPTION]
```

Discrecional

Envíos





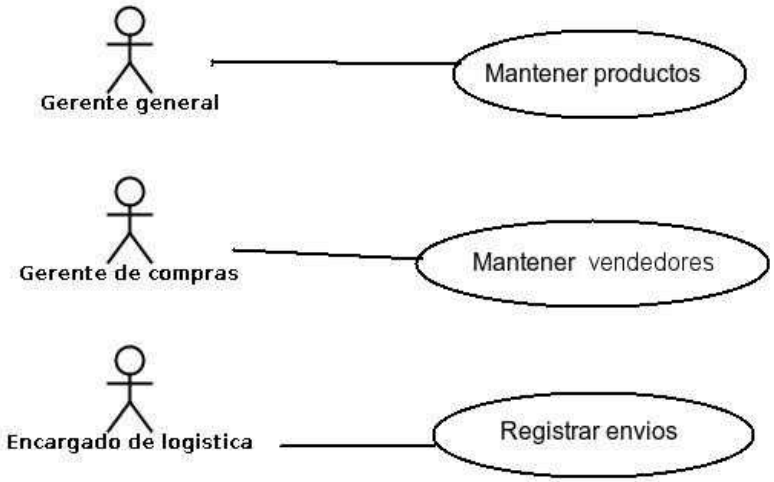
Privilegios por paquetes

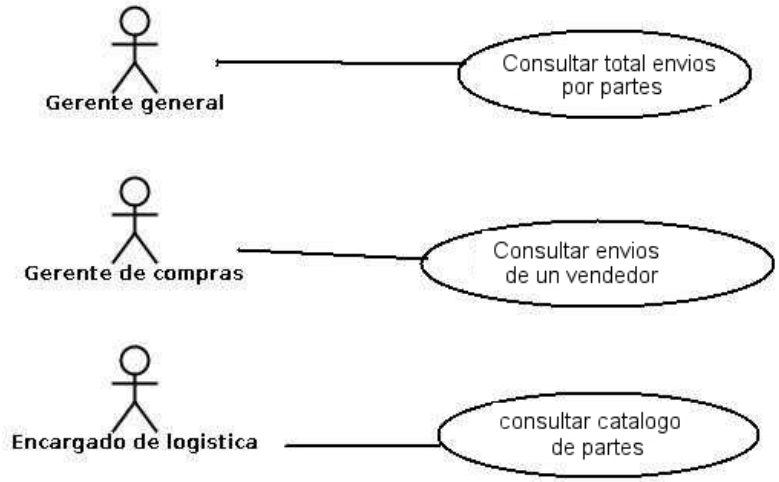
- ¿Cuál serían los paquetes de componentes (CRUD)?

Los únicos datos a modificar son el estado en vendedor y el precio en parte. Las partes no se pueden eliminar.

Discrecional

Envíos





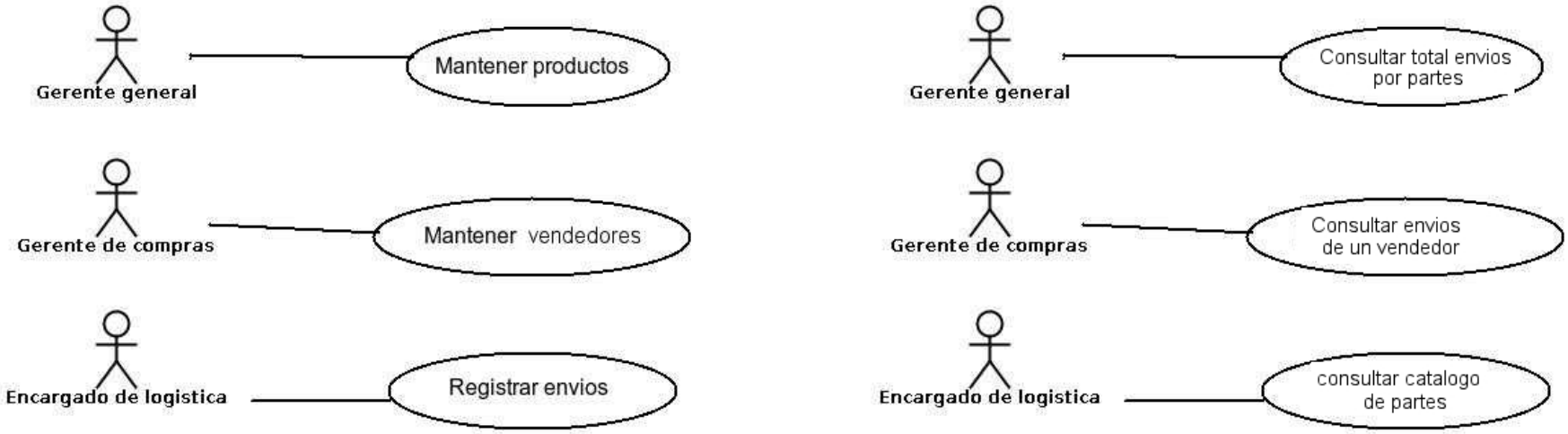
Privilegios por paquetes

- ¿Cuál serían los paquetes de componentes (CRUD)?

  Los únicos datos a modificar son el estado en vendedor y el precio en parte. Las partes no se pueden eliminar.

Discrecional

Envíos



Privilegios por paquetes

- ¿Cuál serían los paquetes de componentes (CRUD)?

  Los únicos datos a modificar son el estado en vendedor y el precio en parte. Las partes no se pueden eliminar.

- ¿Cuál serían los paquetes de seguridad (actores)?

  ¿Cómo quedarían los permisos?

Agenda

Cifrado

Cifrado

Mecanismos

Los caracteres del texto son organizados de una manera diferente

Cifrado

Cifrado

Guardar o transmitir la información sensible de manera cifrada

Mecanismos

- Sustitución

  Se usa una clave de cifrado para determinar el caracter que va a sustituir a cada caracter del texto original

- Permutación

  Los caracteres del texto son organizados de una manera diferente

Agenda

Registro de auditoría

Propósito

Contenido

Registro de auditoría

Propósito

Si hay sospecha, el registro de auditoría permite examinar lo que ha estado sucediendo

:) Verificar que todo está bajo control

: (Para ayudar a señalar dónde hubo un error

Contenido

1. Petición (texto de origen)
2. Terminal desde la que se llamó a la operación
3. Usuario que llamó a la operación
4. Fecha y hora de la operación
5. Varrels, tuplas, atributos afectados
6. Valores antiguos Valores nuevos

Agenda

Discrecional

```
CREATE TABLE VENDEDOR(
  v#          CHAR(2),
  nombre      VARCHAR(20),
  estatus     NUMBER(2),
  ciudad      VARCHAR(10));

CREATE TABLE PARTES(
  p#          CHAR(2),
  parte       VARCHAR(20),
  color       CHAR(10),
  peso        NUMERIC(5,2),
  ciudad      VARCHAR(10));

CREATE TABLE ENVIOS(
  v#          CHAR(2),
  p#          CHAR(2),
  cant        NUMERIC(5));
```

Independiente de valor

```
GRANT SELECT(p#,parte,peso)
ON    PARTES
TO    JUAN,ANA,CARLOS;
```

Discrecional

```
CREATE TABLE VENDEDOR(
  v#          CHAR(2),
  nombre      VARCHAR(20),
  estatus     NUMBER(2),
  ciudad      VARCHAR(10));

CREATE TABLE PARTES(
  p#          CHAR(2),
  parte       VARCHAR(20),
  color       CHAR(10),
  peso        NUMERIC(5,2),
  ciudad      VARCHAR(10));

CREATE TABLE ENVIOS(
  v#          CHAR(2),
  p#          CHAR(2),
  cant        NUMERIC(5));
```

Independiente de valor

Beto es la responsable de la información de las partes puede adicionarlas y consultarlas (todo menos su ciudad) pero no modificarlas ni eliminarlas.

Discrecional

```
CREATE TABLE VENDEDOR(
  v#          CHAR(2),
  nombre      VARCHAR(20),
  estatus     NUMBER(2),
  ciudad      VARCHAR(10));

CREATE TABLE PARTES(
  p#          CHAR(2),
  parte       VARCHAR(20),
  color       CHAR(10),
  peso        NUMERIC(5,2),
  ciudad      VARCHAR(10));

CREATE TABLE ENVIOS(
  v#          CHAR(2),
  p#          CHAR(2),
  cant        NUMERIC(5));
```

Dependiente de valor

```
CREATE VIEW ENVIOS_IMPORTANTES AS
   (SELECT nombre  , parte, cant
    FROM ENVIOS NATURAL JOIN VENDEDORES NATURAL JOIN PARTES
    WHERE estatus > 50);

GRANT SELECT
ON    ENVIOS_IMPORTANTES
TO    LUIS;
```

Discrecional

```
CREATE TABLE VENDEDOR(
  v#          CHAR(2),
  nombre      VARCHAR(20),
  estatus     NUMBER(2),
  ciudad      VARCHAR(10));

CREATE TABLE PARTES(
  p#          CHAR(2),
  parte       VARCHAR(20),
  color       CHAR(10),
  peso        NUMERIC(5,2),
  ciudad      VARCHAR(10));

CREATE TABLE ENVIOS(
  v#          CHAR(2),
  p#          CHAR(2),
  cant        NUMERIC(5));
```

Dependiente de valor

Kate es la responsable de estar pendiente de los envíos de los vendedores de LONDRES. Para eso se le autorizará a únicamente a consultar para cada envío nombre del vendedor, nombre de la parte y cantidad.

Discrecional

```
CREATE TABLE VENDEDOR(
  v#          CHAR(2),
  nombre      VARCHAR(20),
  estatus     NUMBER(2),
  ciudad      VARCHAR(10));

CREATE TABLE PARTES(
  p#          CHAR(2),
  parte       VARCHAR(20),
  color       CHAR(10),
  peso        NUMERIC(5,2),
  ciudad      VARCHAR(10));

CREATE TABLE ENVIOS(
  v#          CHAR(2),
  p#          CHAR(2),
  cant        NUMERIC(5));
```

Resumen estadístico

```
CREATE VIEW RESUMEN_ENVIOS AS
    (SELECT p#, SUM(cant) AS totales
     FROM ENVIOS
     GROUP BY p#);
```

```
GRANT SELECT
ON RESUMEN_ENVIOS
TO FIDEL;
```

Discrecional

```
CREATE TABLE VENDEDOR(
  v#          CHAR(2),
  nombre      VARCHAR(20),
  estatus     NUMBER(2),
  ciudad      VARCHAR(10));

CREATE TABLE PARTES(
  p#          CHAR(2),
  parte       VARCHAR(20),
  color       CHAR(10),
  peso        NUMERIC(5,2),
  ciudad      VARCHAR(10));

CREATE TABLE ENVIOS(
  v#          CHAR(2),
  p#          CHAR(2),
  cant        NUMERIC(5));
```

Resumen estadístico

A los socios de la empresa les interesa conocer el número de vendedores con que cuentan en cada ciudad.

Discrecional

```
CREATE TABLE VENDEDOR(
  v#          CHAR(2),
  nombre      VARCHAR(20),
  estatus     NUMBER(2),
  ciudad      VARCHAR(10));

CREATE TABLE PARTES(
  p#          CHAR(2),
  parte       VARCHAR(20),
  color       CHAR(10),
  peso        NUMERIC(5,2),
  ciudad      VARCHAR(10));

CREATE TABLE ENVIOS(
  v#          CHAR(2),
  p#          CHAR(2),
  cant        NUMERIC(5));
```

Dependiente de contexto

```
CREATE VIEW HORAS_OFICINA AS
    (SELECT *
     FROM  VENDEDORES
     WHERE '08' <= TO_CHAR(SYSDATE,'HH24')
       AND TO_CHAR(SYSDATE,'HH24')<='16'
       AND TO_CHAR(SYSDATE,'DY') NOT IN ('SAT','SUN'));
```

```
GRANT SELECT
ON    HORAS_OFICINA
TO    CONTABILIDAD;
```

Discrecional

```
CREATE TABLE VENDEDOR(
  v#          CHAR(2),
  nombre      VARCHAR(20),
  estatus     NUMBER(2),
  ciudad      VARCHAR(10));

CREATE TABLE PARTES(
  p#          CHAR(2),
  parte       VARCHAR(20),
  color       CHAR(10),
  peso        NUMERIC(5,2),
  ciudad      VARCHAR(10));

CREATE TABLE ENVIOS(
  v#          CHAR(2),
  p#          CHAR(2),
  cant        NUMERIC(5));
```

Dependiente de contexto

Martha es la responsable de mantener la información de los proveedores. Ella puede adicionarlos, modificarlos (pero sólo los lunes) y consultar todo menos su estatus. No puede eliminarlos.