

# Bases de dades

## Documentació

PAULA VERA  
MARCOS SUÁREZ  
GESTOR D'INCIDÈNCIES

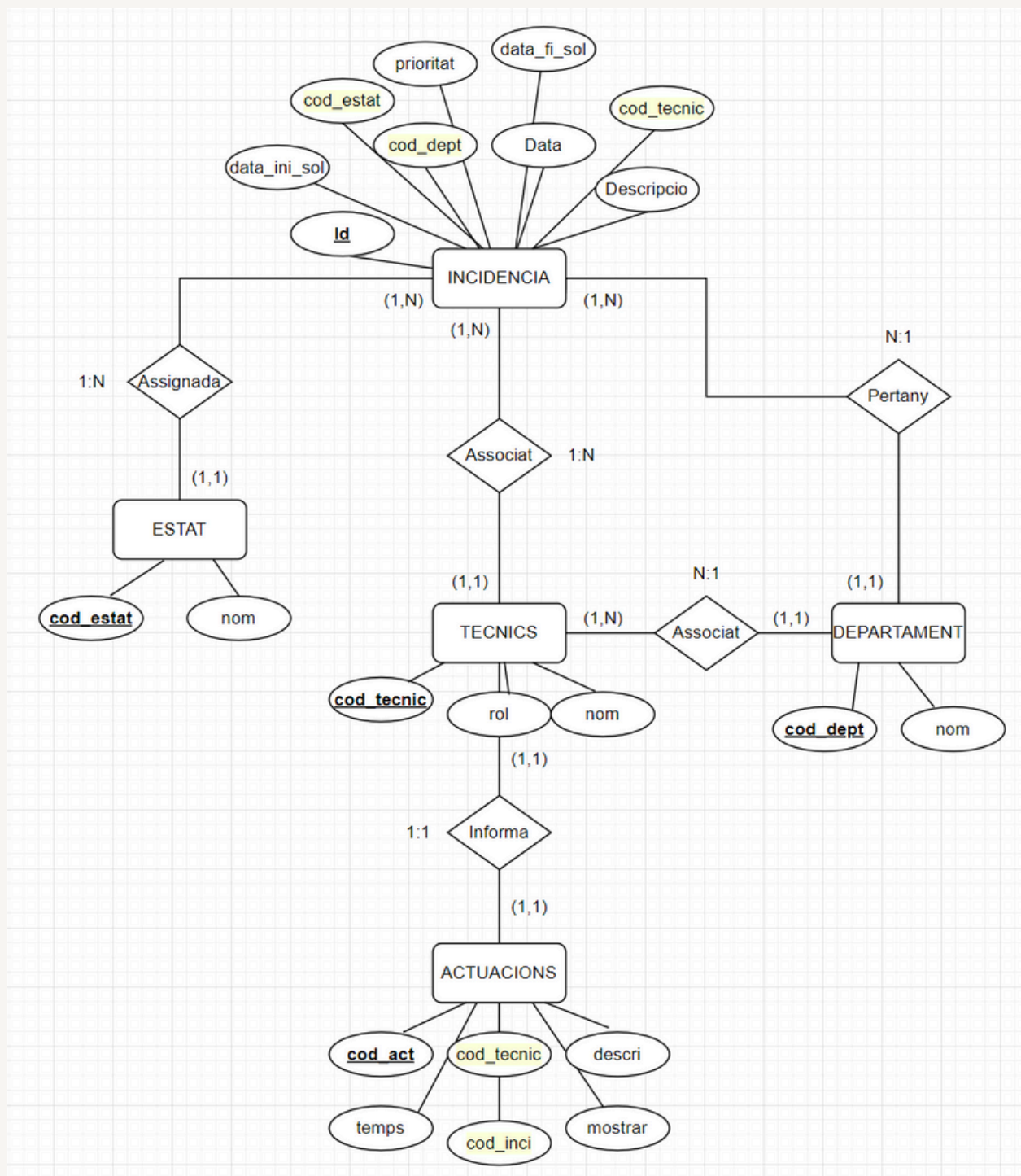
# Index

01 DIAGRAMA E-R.....	2
02 Documentació de les consultes principals.....	3-7

# **01**

## **Diagrama E-R**

## Diagrama E-R



**02**

# **Documentació de les consultes principals**

## Documentació de les consultes principals

Hem agafat els enunciats i afegirem una captura de les nostres consultes per respondre a aquestes peticions.

- *Registrar nova incidència: Pantalla dirigida als usuaris, permet registrar una nova incidència. L'usuari haurà d'introduir únicament el departament al que pertany, la data de la incidència (no fa falta especificarla i el sistema la pot agafar automàticament) i una descripció curta de la incidència, no farà falta indicar el id de la incidència (serà autonumeric)*

```
function guardarActu($conn, $id, $idincidencia, $descripcio, $visible, $temps, $data) {
    $sql = "INSERT INTO ACTUACIONS (cod_tecnic, cod_inci, descri, mostrar, temps, `data`) VALUES (?, ?, ?, ?, ?, ?)";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("iisiis", $id, $idincidencia, $descripcio, $visible, $temps, $data);

    if (!$stmt->execute()) {
        die("Error en la consulta: " . $stmt->error);
    }

    $stmt->close();
}
```

- *Estat de una incidència: Es pregunta el codi d'una incidència i per ella es mostraran totes les actuacions (les que siguin visibles pel usuari) ordenades cronològicament.*

```
function llegirActuUsu($conn, $idincidencia) {
    $sql = "SELECT cod_inci, descri, temps, `data`
    FROM ACTUACIONS
    WHERE cod_inci = ? AND mostrar = 1 -- (1 equival a si)
    ORDER BY `data` DESC";

    $stmt = $conn->prepare($sql);
    $stmt->bind_param("i", $idincidencia);

    if (!$stmt->execute()) {
        die("Error executing statement: " . $stmt->error);
    }
}
```

- *Modificar incidència: Aquesta pantalla mostra un llistat d'incidències no resoltes i el responsable d'informàtica escull la incidència que vol i assignarà cada una de les noves incidències a un tècnic i assignarà una prioritat. Únicament s'han de mostrar les incidències no resoltes.*

```
function llegirIncidenciesAdminist($conn, $id) {
    $sql = "SELECT Id, cod_dept, Descripcio, Data, cod_estat, cod_tecnic, prioritat FROM INCIDENCIA WHERE Id = $id
    AND cod_estat IN('1','2') --això equival als estats que no son 'solucionat'
    ORDER BY `data` DESC";
    $stmt = $conn->prepare($sql);

    if (!$stmt->execute()) {
        die("Error executing statement: " . $stmt->error);
    }
}
```

Per la part de modificar, un exemple amb els estats:

```
function actualitzarEstat($conn, $id, $estat) {
    $sql = "UPDATE INCIDENCIA SET cod_estat = ? WHERE Id = ?";
    $stmt = $conn->prepare($sql);

    if (!$stmt) {
        die("Error en preparar la consulta: " . $conn->error);
    }
}
```

- *Registrar actuació:* A la pantalla s'arriba amb un nombre d'incidència i es mostren totes les actuacions fins el moment. Permet introduir dades d'una nova actuació, el temps que s'ha trigat, s'introdueix la descripció i s'indica si la descripció esta visible o no per al usuari. Des d'aquest formulari també serà possible indicar la finalització de la incidència, així com registrar la data de la finalització.

```
function guardarActu($conn, $id, $idincidencia, $descripcio, $visible, $temps, $data) {
    $sql = "INSERT INTO ACTUACIONS (cod_tecnic, cod_inci, descri, mostrar, temps, `data`) VALUES (?, ?, ?, ?, ?, ?)";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("iisiis", $id, $idincidencia, $descripcio, $visible, $temps, $data);

    if (!$stmt->execute()) {
        die("Error en la consulta: " . $stmt->error);
    }

    $stmt->close();
}
```

Afegim la captura de la crida a les funcions que es demanen per no repetir captures amb consultes semblants:

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {

    if (isset($_POST['submit_actuacio'])) {

        if (!empty($descripcio)) {
            guardarActu($conn, $id, $idincidencia, $descripcio, $visible, $temps,$data);
        }
    }
    if (isset($_POST['submit_estat'])) {

        if (!empty($estat)) {
            actualitzarEstat($conn, $idincidencia, $estat);
        }
        if (!empty($datafi)) {
            dataFiAct($conn, $idincidencia, $datafi);
        }
    }
}
llegirActu($conn, $idincidencia);
}
```

- *Informe de Tècnics: S'ha de mostrar per cada tècnic totes les incidències que encara no estan tancades. Concretament, es vol saber la data d'inici de la incidència i el total de temps dedicat fins al moment.. Les incidències per cada tècnic s'han d'agrupar per prioritat (Alta - Mitjana - Baixa). HEU DE FER UNA JOIN de 3 TAULES i AGRUPAR PER INCIDÈNCIA SUMANT EL TEMPS DE CADA ACTUACIÓ*

```
function llegirInformeTec($conn) {  
    $sql = "SELECT  
        I.Id AS ID_Incidencia,  
        T.rol AS Tècnic,  
        T.nom AS Nom,  
        I.prioritat AS Prioritat,  
        I.Data AS Data_Inici,  
        SUM(A.temps) AS Temps_Total  
    FROM  
        INCIDENCIA I  
    INNER JOIN  
        TECNICS T ON I.cod_tecnic = T.cod_tecnic  
    INNER JOIN  
        ACTUACIONS A ON I.Id = A.cod_inci  
    INNER JOIN  
        ESTAT E ON I.cod_estat = E.cod_estat  
    WHERE  
        E.nom != 'Solucionada'  
    GROUP BY  
        I.Id, T.rol, T.nom, I.Data, I.prioritat  
    ORDER BY  
        T.rol, I.prioritat;  
    ";  
  
    $stmt = $conn->prepare($sql);  
}
```



- *Consum per departaments: Es tracta d'un informe en el que es mostrarà per cada departament de la empresa, el nombre d'incidències que han reportat i el temps total dedicat a elles. COMPLICAT (ÉS NECESSARI FER SERVIR SUBCONSULTES)*

```
function llegirDeptInfo($conn) {
    $sql = "SELECT
        D.nom AS Departament,
        (SELECT COUNT(*)
         FROM INCIDENCIA I
         WHERE I.cod_dept = D.cod_dept
        ) AS Nombre_Incidencies,
        (SELECT SUM(A.temps)
         FROM INCIDENCIA Inc
         JOIN ACTUACIONS A ON A.cod_inci = Inc.Id
         WHERE Inc.cod_dept = D.cod_dept
        ) AS Temps_Total
    FROM DEPARTAMENT D;";
    $stmt = $conn->prepare($sql);
```

- *Estadístiques d'Accés: Panell que mostra informació estadística sobre els accessos a l'aplicació. Es mostrarà un resum amb el nombre d'accessos totals, les pàgines més visitades i els usuaris més actius. També permetrà filtrar els logs per data, usuari i pàgina visitada. Les dades es mostraran en forma de taules i també es proporcionaran gràfics per visualitzar tendències d'ús de l'aplicació al llarg del temps. Aquesta informació estarà disponible només pels usuaris amb rol d'administrador. NECESSARI IMPLEMENTAR AGREGACIONS DE MONGODB*

```
function guardarLog($collection, $User, $dataUsuari, $ipUsuari, $pageUsuari, $user_agent)
{
    $collection->insertOne([
        'User' => $User,
        'Data' => $dataUsuari,
        'ip_origin' => $ipUsuari,
        'pagina visitada' => $pageUsuari,
        'navegador' => $user_agent
    ]);
}

function llegirDadesMongodb($collection)
{
    $documents = $collection->find();

    echo "<div class='container mt-4'>";
    echo "<h1> Dades MongoDB del gestor </h1>";
    echo "<div class='p-3 bg-light border rounded'>";
    echo "<table class='table table-striped'>";
    echo "<tr><th>Usuari</th><th>Data</th><th>Pàgina visitada</th><th>Navegador</th></tr>";

    foreach ($documents as $document) {
        echo "<tr>";
        echo "<td>" . htmlspecialchars($document['User'] ?? "x") . "</td>";
        echo "<td>" . htmlspecialchars($document['Data'] ?? "x") . "</td>";
        echo "<td>" . htmlspecialchars($document['pagina visitada'] ?? "x") . "</td>";
        echo "<td>" . htmlspecialchars($document['navegador'] ?? "x") . "</td>";
        echo "</tr>";
    }
}
```