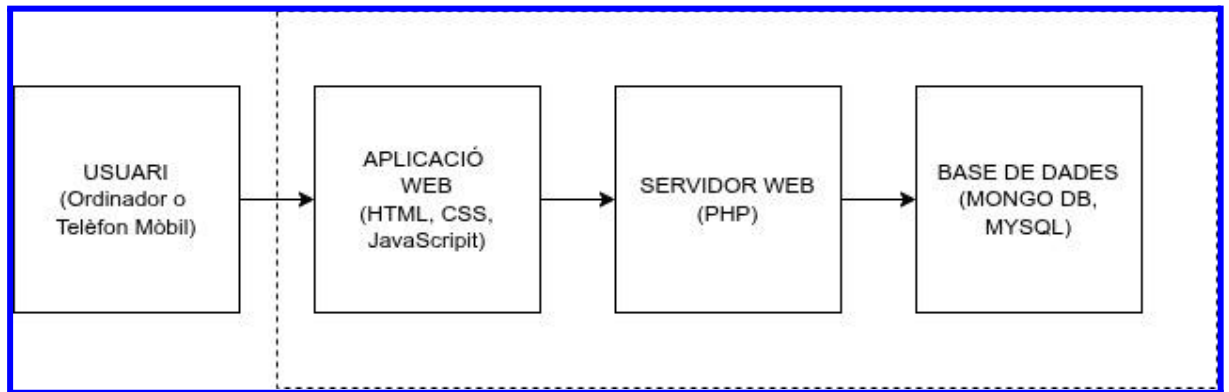


# SISTEMES DOCUMENTACIÓ

## Descripció del Projecte:

### • Diagrama de l'arquitectura del sistema



### 1. Gestió d'Incidències Biel & Amin

- Aquest projecte consisteix en una aplicació web per gestionar **les incidències tècniques dins d'una organització**. L'objectiu principal és **facilitar la comunicació entre usuaris, tècnics i administradors**, assegurant que cada problema es resolgui de manera eficient.

### 2. Tipus d'Usuari

- L'aplicació està dissenyada perquè diferents tipus d'usuaris puguin interactuar amb el sistema. Cada rol té accés a diferents funcionalitats:
  - a) Els usuaris són els treballadors de l'organització que poden **registrar incidències** quan tenen un problema tècnic.
  - **Funcionalitats: Crear una incidència** → Descriure el problema i associar-lo a un departament. **Consultar l'estat de la seva incidència** → Veure si està pendent, en procés o resolta.
  - b) Els tècnics són els responsables de **solucionar les incidències**, donant suport als usuaris.
  - **Funcionalitats: Veure les incidències assignades** → Gestionar les tasques pendents. **Registrar una actuació** → Informar dels passos realitzats per solucionar el problema. **Tancar incidències** → Confirmar que el problema s'ha resolt satisfactòriament.

# SISTEMES DOCUMENTACIÓ

- c) Els administradors **supervisen el sistema, gestionen les incidències i analitzen dades** per millorar la gestió.
- **Funcionalitats: Consultar totes les incidències** → Veure les dades i estat de cada problema. **Modificar incidències** → Canviar tipus, assignar tècnics i establir prioritats. **Generar informes** → Accedir a estadístiques detallades sobre la resolució d'incidències.

## 3. Documentació *docker-compose.yml*

- Aquest fitxer docker-compose.yml defineix i gestiona diversos contenidors Docker per desplegar el sistema de gestió d'incidències.
- Els serveis principals són:
  - a) **Aplicació Web amb PHP i Apache** (apache\_php\_mysql\_mongodb)
  - b) **Base de dades MySQL** (db)
  - c) **Adminer (Interfície web per gestionar MySQL)** (adminer)
  - d) **Base de dades NoSQL MongoDB** (mongo)
  - e) **Interfície de gestió Mongo Express** (mongo-express)
- a) **Aquest contenidor executa l'aplicació web amb PHP i Apache.** Carrega automàticament les dependències de composer. Exposa el port **8080**, accessible des del navegador. **Munta el directori php/ a /var/www/html**, permetent que el codi PHP es desplegui dins del contenidor.

```
services:
  apache_php_mysql_mongodb:
    #La imatge php:apache no té tots els drivers. hem de fer una imatge a mida
    # Podeu fer servir el dockerfile que hi ha a /images i fer el build
    # o podeu agafar aquest imatge, ebota/daw:apache_php_mysql_mongodb,
    # que ja està construïda i està a dockerhub. Teniu més info a:
    # https://hub.docker.com/r/ebota/daw
    image: ebota/daw:apache_php_mysql_mongodb
    ports:
      - "8080:80"
    volumes:
      - ./php:/var/www/html
    # Es necessari que hi hagi el fitxer composer.json a la carpeta php i la següent directiva
    # perquè el php instal·li les dependències de mongo
    command: /bin/bash -c "composer install --no-interaction --no-plugins --no-scripts && echo WEBSERVER UP AND"
    environment:
      VAR1: "sóc una pera i estic al docker-compose"
      VAR2: ${VAR2}
    depends_on:
      - db
```

# SISTEMES DOCUMENTACIÓ

- b) **Contenidor que executa MySQL per emmagatzemar incidències, tècnics i usuaris.** Les credencials estan definides a environment. Les dades de la BD es guarden en **volums persistents (db\_data)** per evitar pèrdues. No exposa ports externs **per seguretat**, ja que l'aplicació web accedeix internament.

```
db:
# Tota la informació d'aquesta imatge està https://dockerhub.com/\_/mysql
image: mysql:9.3
environment:
  MYSQL_ROOT_PASSWORD: passwordDeRootQueNoShaDeFerServirMai
  # És millor no crear la BBDD aquí, ja que no pots controlar la codificació
  # de caràcters i aleshores donarà problemes amb accents i caràcters especials
  # La BBDD es crearà a l'inici del contenidor amb els script d'inicialització
  # MYSQL_DATABASE: a24biedommar_ProjecteFinal_MySql
  MYSQL_USER: usuari
  MYSQL_PASSWORD: paraula_de_pas
  LANG: C.UTF-8
# El mysql no s'exposa a l'exterior
# L'aplicació web hi accedirà per la xarxa interna de docker
# ports:
#   - "3306:3306"

# La carpeta de mysql ha d'estar al .gitignore per no pujar-la al repositori
volumes:
  - ./db_data:/var/lib/mysql
  - ./db_init:/docker-entrypoint-initdb.d
```

- c) **Aquest contenidor és una interfície web per accedir i administrar la base de dades MySQL.** Accessible via <http://localhost:8081>. Utilitza les mateixes credencials de MySQL per iniciar sessió.

```
adminer:
  image: adminer
  ports:
    - "8081:8080"
  depends_on:
    - db
```

# SISTEMES DOCUMENTACIÓ

- d) **Contenidor que executa MongoDB per gestionar dades no relacionals.** Exposa el port **27017**, utilitzat per connexions internes. Guarda les dades en un **volum persistent** (mongodb\_data) per evitar pèrdues.

```
mongo:
  image: mongo:latest
  volumes:
    #Named volume, no cal posar res al .gitignore
    #perque no es crea una carpeta al host sinó que es crea un volum a docker
    # i es pot veure amb docker volume ls
    # i es pot eliminar amb docker volume rm nom_del_volum
    - mongodb_data:/data/db
  ports:
    - "27017:27017"
  environment:
    MONGO_INITDB_ROOT_USERNAME: root
    MONGO_INITDB_ROOT_PASSWORD: example
```

- e) **Aquest contenidor proporciona una interfície web per gestionar MongoDB.** Accessible via <http://localhost:8082>. Permet veure i administrar les col·leccions de la BD NoSQL.

```
mongo-express:
  image: mongo-express
  ports:
    - "8082:8081"
  environment:
    #Per entrar a la web de mongo-express, les credencials són
    #admin
    #pass
    ME_CONFIG_MONGODB_URL: mongodb://root:example@mongo:27017/
  depends_on:
    - mongo
```

# SISTEMES DOCUMENTACIÓ

- f) **Per evitar pèrdues de dades quan es reinicien els contenidors, les bases de dades usen volums Docker. MySQL → db\_data MongoDB → mongodb\_data**

```
volumes:  
  mongodb_data:
```

## 4. Base de Dades

- **El sistema utilitza una base de dades relacional (MySQL) per gestionar les incidències i els usuaris.**
- Taules principals i relacions
- La base de dades conté les següents taules:
  - **Usuari** → Registra els usuaris que poden crear incidències.
  - **Incidència** → Guarda la informació dels problemes reportats.
  - **Prioritat** → Defineix la urgència de cada incidència (Baixa, Mitjana, Alta, Crítica).
  - **Tipus d'incidència** → Categoritza les incidències (Software, Connexió, Hardware, etc.).
  - **Estat** → Indica si una incidència està pendent, en procés o resolta.
  - **Tècnic** → Assigna un professional per resoldre incidències.
  - **Departament** → Cada incidència pertany a un departament específic.
  - **Actuació** → Registra les accions fetes pels tècnics per solucionar una incidència.

# SISTEMES DOCUMENTACIÓ

## 5. Manteniment del Sistema

- **Per garantir el bon funcionament de l'aplicació, cal optimitzar la gestió de recursos i prevenir errors.**
- Optimització del sistema
  - a) **Monitorització dels contenidors Docker** → Revisar l'estat amb docker ps.
  - b) **Gestió de logs** → Analitzar errors amb docker logs (i el nom).
- Resolució de problemes comuns
  - a) **Si l'aplicació no respon, aleshores s'ha de** comprovar que docker-compose up ha iniciat correctament. I Revisar els logs per detectar possibles errors (docker logs apache\_php\_mysql\_mongodb).
  - b) Si hi ha **error en la connexió MySQL** es verifica si el contenidor db està en marxa (docker ps). I confirma que les credencials de connexio.php són correctes.

## 6. Còpies de Seguretat amb Docker

- **Les còpies de seguretat són una mesura essencial per garantir la protecció i recuperació de dades en cas d'errors o pèrdues inesperades.** En el sistema de gestió d'incidències, és fonamental disposar d'un mecanisme per guardar regularment la informació de la base de dades.
- **Importància de les còpies de seguretat**
- Les còpies de seguretat permeten restaurar la base de dades a un estat anterior en cas de fallades del sistema, errors humans o problemes amb el servidor. Aquest procés és crucial per **evitar la pèrdua d'informació sobre incidències, tècnics, usuaris i altres elements del sistema.**
- **Estratègia de còpies de seguretat**
- Per garantir la seguretat de les dades, es recomana seguir **un esquema de backup automatitzat.** Això inclou:
  - **Còpies diàries** per assegurar que sempre es pot restaurar una versió recent de la base de dades.
  - **Còpies externes** en un servidor segur per evitar pèrdua de dades en cas de problemes amb el sistema principal.

# SISTEMES DOCUMENTACIÓ

- **Manteniment dels backups** en diversos formats per garantir compatibilitat i facilitat de restauració.
- **Restauració de dades**
- En cas d'una fallada del sistema, es pot restaurar la base de dades mitjançant **una còpia de seguretat recent**. Aquest procés permet recuperar **totes les incidències i la informació associada**, evitant la interrupció del servei per als usuaris i tècnics.