Resum de la plataforma PHP

Taula de continguts

- 1. Servidor web
- 2. Versió de PHP
- 3. Plataformes LAMP
- 4. Connexions HTTP
- 5. Bases de dades
    - 5.1. MySQL/MariaDB
    - 5.2. Oracle
    - 5.3. MongoDB
- 6. Altres extensions/configuracions
- 7. Debug
- 8. Logs d'aplicació
- 9. Gestor de continguts (Drupal)
- 10. Framework de Desenvolupament (Symfony)
- 11. Frameworks JS
- 12. Desenvolupament d'APIs
- 13. Control de versions
- 14. Disseny i maquetació
- 15. Gestió d'Usuaris
- 16. Gestió de dependències
- 17. Alta disponibilitat
    - 17.1. Desplegament del codi
    - 17.2. Persistència de les sessions
    - 17.3. Bases de dades
    - 17.4. Com atacar un node concret localment
- 18. Anàlisi estàtic del codi (SonarQube)
- 19. Seguretat
- 20. Empreses

Tecnologies, llibreries i frameworks principals de la Plataforma PHP.

1. Servidor web

Apache/2.4.6 (Red Hat)

2. Versió de PHP

La versió actual de PHP a explotació és la 8.4.17 i és sobre la que desenvolupem els nous projectes.

3. Plataformes LAMP

- LAMP 1: PHP 5.2 (extingida)
- LAMP 2: PHP 5.3 (en procés d'extinció)
- LAMP 3: PHP 7.4 (en procés d'extinció)

- LAMP 3 HA: PHP 7.4 (alta disponibilitat, en producció) - 2 nodes PHP + 3 nodes MariaDB + 3 nodes MongoDB
- LAMP 4: PHP 8.4 (alta disponibilitat, en beta) - 2 nodes PHP + 3 nodes MariaDB + 3 nodes MongoDB

4. Connexions HTTP

Per a realitzar connexions HTTP o connectar-se amb serveis web REST s'utilitzarà la llibreria integrada cURL, no utilitzarem crides tipus file_get_contents o stream_socket_client o similars per connectar-se a una URL sempre que sigui possible.

5. Bases de dades

5.1. MySQL/MariaDB

La versió actual del MySQL (LAMP 3) és la 5.7
La versió actual del MySQL/MariaDB (LAMP 3HA i LAMP4) és la 11.4. La base de dades són tres nodes en alta disponibilitat que cal configurar a la connexió de l'aplicació.
La taula de compatibilitats amb Mysql es pot consultar aquí: https://mariadb.com/kb/en/mariadb-vs-mysql-compatibility/ [1]

5.2. Oracle

La versió actual d'Oracle és la 11g, hi ha un oracle corporatiu amb UTF-8 i un altre amb la codificació especial WE8ISO8859P1.

Utilitzarem preferentment el driver natiu OCI8. Si treballem amb l'Oracle codificat amb WE8ISO8859P1 per a la correcte gestió dels caràcters especials especificarem el charset WE8ISO8859P1 en el moment de la connexió i un cop recuperades les dades convertirem la informació de ISO-8859-1 a UTF-8 per poder treballar normalment amb UTF-8. Si no es fa així s'han detectat diferents incidències de codificació de caràcters o de recompte del nombre de caràcters.

Exemple connexió a mida:

```
$dbh = oci_connect($this->username, $this->password, $this->host, 'WE8ISO8859P1');
$sth = oci_parse($dbh, 'SET ROLE ALL');
oci_execute($sth);
$sth = oci_parse($dbh, 'SELECT 1 FROM DUAL');
oci_execute($sth);

// Etc...

$data = iconv('ISO-8859-1', 'UTF-8', $data);
```

On el host pot ser, per exemple: $this->host = 'psi.world';

En el cas de symfony treballarem amb Doctrine com a ORM sempre que sigui possible.

5.3. MongoDB

Com a base de dades NonSQL disposem de MongoDB 5 per a treballar amb projectes Non-sql.

- DEV:
  - su0747.corpo.ad.diba.es:27017,su0748.corpo.ad.diba.es:27017,su0749.corpo.ad.diba.es:27017
  - Replicaset: dibarsdev
- PRD:

- su0798.corpo.ad.diba.es:27017,su0799.corpo.ad.diba.es:27017,su0800.corpo.ad.diba.es:27017
- Replicaset: dibars

6. Altres extensions/configuracions

- Extensió PHP Mbstring com a biblioteca unicode.
- Registres globals de PHP SEMPRE inactius.
- PHP short tags desactivat
- Suport per OpenSSL actiu.
- cURL actiu per a gestió de peticions HTTP.
- Gestió d'imatges i generació de miniatures amb GD 2.1.0
- Suport amb gettext actiu.
- Altres llibreries activades: bz2, iconv, intl, zlib, wkhtmltopdf (Lamp4)

7. Debug

A l'entorn de desenvolupament s'utilitza l'error reporting E_ALL i XDebug per depurar i analitzar el codi que s'executa.

En el cas de Symfony l'entorn de desenvolupament mostrarà sempre la barra de debug de Symfony i l'entorn de producció no.

8. Logs d'aplicació

Els logs de les aplicacions han de ser rotatius, no poden créixer indefinidament. Per defecte rotarem cada 30 dies, per exemple si el projecte utilitza monolog, com pot ser amb Symfony configurarem rotating_file:

```
monolog:
    handlers:
        main:
            type:  rotating_file
            path:  '%kernel.logs_dir%/%kernel.environment%.log'
            level: debug
            max_files: 30
```

9. Gestor de continguts (Drupal)

El nostre gestor de continguts és **Drupal**, actualment Drupal 11 per a nous projectes.

No acceptem projectes PHP amb altres gestors de continguts com Wordpress o Joomla.

Tot nou projecte ha de tenir el seu propi domini o subdomini.

Pel nucli de Drupal s'utilitzarà MySQL/MariaDB i si cal interconnexió amb dades provinents de la base de dades Oracle corporativa es farà a través de serveis web o programant un mòdul a mida que es connectarà l'Oracle corporatiu.

Tota instal·lació de drupal **sempre** utilitzarà un **PREFIX** a les seves taules definit a la connexió de settings.local.php, i **MAI** instal·larem sense prefix.

L'usuari uid = 1 sempre serà el superadminsitrador, es dirà **admindiba** i tindrà un correu de la Diputació (informàtica). Per motius de seguretat no existirà en el sistema cap usuari que es digui: admin, test o administrator.

Hi ha documentació més detallada de com treballar amb drupal aquí:
http://comunitatdstsc.diba.cat/wiki/desenvolupament-corporatiu-amb-drupal [2]

10. Framework de Desenvolupament (Symfony)

El nostre framework de desenvolupament PHP és **Symfony** i es desenvolupa preferentment amb **versions LTS**.

No acceptem projectes PHP amb altres frameworks com Laravel si no s'inclou el manteniment de **TOTA** la vida del projecte en un contracte ja que només podem oferir suport amb recursos interns per projectes Drupal o Symfony.

Els nous projectes han d'utilitzar la darrera versió estable LTS (Long Term Support) actualment Symfony 7.4 LTS (05/02/26).

Les versions LTS tenen 36 mesos de suport oficial i les normals només 8 mesos de suport oficial. Poseu consultar els versions i la duració del suport aquí: https://symfony.com/releases [3]

Per cada projecte a mida s'utilitza el que s'anomena un "sandbox", on cada projecte està auto-contingut, és completament autònom i disposa de la seva pròpia versió de Symfony.

Per als projectes nous es recomana la programació de proves unitàries i funcionals utilitzant el framework de symfony i PHPUnit amb un bon "code coverage" sobre les funcionalitats desenvolupades.

Segons la naturalesa i les necessitats del projecte s'escollirà la base de dades més adequada MariaDB/MySQL o Oracle o MongoDB.

Tots els projectes Symfony utilitzaran tres entorns (dev, pre i prod) que hauran d'estar correctament separats a la configuració en fitxer .env

Important!: La configuració de la base de dades estarà definida **SEMPRE** com un paràmetre d'entorn.

Cada entorn te la seva configuració de base de dades, el seu host i el seu usuari/password.

11. Frameworks JS

Es contempla el desenvolupament de projectes amb frameworks JS com Angular, React o Vue.

Aquest tipus de desenvolupament sempre inclouran un fitxer config.json amb les variables que han de canviar en cada entorn com URLs de crides a APIs externes, la url de base del projecte, etc. No s'acceptaran projectes compilats amb les variables d'entorn incrustades dins de la compilació. Cal poder desplegar el projecte mimificat en tots els entorn modificant només els valors de config.json

12. Desenvolupament d'APIs

L'eina preferent per a desenvolupar APIs REST amb PHP és: API Platform [4]

13. Control de versions

S'utilitzarà el Git corporatiu com a sistema de control de versions del programari.

14. Disseny i maquetació

El disseny i la maquetació es farà amb **Bootstrap** amb la última versió de Bootstrap disponible per la maqueta corporativa que subministrarà la Diputació de Barcelona agafant com a referència:

- Maqueta corporativa [5]

15. Gestió d'Usuaris

La gestió d'usuaris i validació d'usuaris interns de la casa es farà a través del sistema de gestió d'usuaris corporatiu VUS. Hi ha un mòdul fet a mida d'integració amb el VUS per a Drupal i exemples d'ús del servei web del VUS amb Symfony.

16. Gestió de dependències

Tant en els projectes Drupal com Symfony s'utilitza composer [6] per gestionar les versions i les dependències dels seus components.

Ha de ser possible actualitzar els components d'un projecte via: composer update --with-dependencies --prefer-dist

17. Alta disponibilitat

La LAMP3HA en alta disponibilitat requereix algunes consideracions addicionals a tenir en compte. En ser un servei en alta disponibilitat cal tenir en compte que ja no tenim una única màquina sinó vàries, i en funció de l'aplicació caldrà realitzar configuracions per garantir el correcte funcionament de l'aplicació.

Actualment tenim per a cada entorn:

- 2 nodes PHP
- 3 nodes MySQL/MariaDB
- 3 nodes MongoDB

17.1. Desplegament del codi

El desplegament del codi es fa a través de scripts sh pel que és imprescindible que tot projecte tingui el codi sincronitzat al repositori de codi corporatiu GIT i disposi d'un script sh per desplegar a cada un dels entorns.

Podeu trobar més informació amb exemples de scripts de desplegament en el següent enllaç: Desplegaments PHP [7]

Llista d'entorns i servidors PHP:

- LAMP3
  - Desenvolupament: su0023
  - Producció: su0034
- LAMP3HA
  - Desenvolupament: su0245(node1), su0246(node2)
  - Pre-Producció: su0815(node1), su0816(node2)
  - Producció: su0253(node1), su0254(node2)
- LAMP4
  - Desenvolupament: su0987(node1), ?(node2)

17.2. Persistència de les sessions

Com que tenim 2 nodes PHP, si la nostre aplicació fa ús de les sessions de l'usuari utilitzem les **sticky sessions**, on el mateix sistema un cop un usuari accedeix a un dels dos nodes aleatoris la resta de peticions es mantenen al mateix node durant tota la sessió.

Per a les sticky sessions no cal configurar res, aquest es el comportament per defecte de la plataforma. Si no es requeix una sticky session perquè no es fa ús de les sessions es pot demanar un accés totalment aleatori pel projecte segons la càrrega de cada node.

17.3. Bases de dades

Tenim 3 nodes MariaDB i per fer el balanceig utilitzem el maxScale pel que possarem la IP del maxScale a la nostre connexió:

- Desenvolupament: su0247, su0248, su0249 <== maxScale IP: 192.168.156.88:3307
- Preproduccio: su0834, su0835, su0836 <== maxScale IP: 192.168.156.88:4307
- Producció: su0255, su0256, su0257 <== maxScale IP: 192.168.57.105:3307

Com que tenim 3 nodes MongoDB cal configurar els res servidors a la connexió:

- Desenvolupament: mongodb://su0074.corpo.ad.diba.es:27017,su0075.corpo.ad.diba.es:27017,su0076.corpo.ad.diba.es:27017
- Producció: mongodb://mongodb01:27017,mongodb02:27017,mongodb03:27017

Si ens connectem a un únic node no estarem treballant en alta disponibilitat.

17.4. Com atacar un node concret localment

Quan ens trobem desenvolupant codi en aquest entorn, ens podem trobar en la situació de caure en un node on el codi no es desplega de forma directa, sinó que es copia d'un altre node cada x minuts. Això pot convertir la tasca de provar el codi en una de molt feixuga.
Per a evitar casos com aquests, podem atacar un node en concret de la següent manera:

1) Obrirem un editor en mode administrador.
2) Obrirem l'arxiu ubicat a "C:\Windows\System32\drivers\etc\hosts".
3) Afegirem una línia seguint el següent format: [IP del node] [Domini]
4) Sobreescriure el fitxer.

Si ho hem fet correctament, els navegadors que utilitzem en el nostre ordinador redireccionaran [Domini] cap a [IP del node].
IP DVOL3HA: 192.168.156.175 [8] (master)

18. Anàlisi estàtic del codi (SonarQube)

Tot el codi fet a mida es passarà pel SonarQube corporatiu de la Diputació de Barcelona per fer-ne un anàlisi estàtic del codi.

Es proporcionarà als desenvolupadors un usuari on revisar els informes del codi del SonarQube.

Caldrà resoldre les deteccions automàtiques del sonarQube marcades com a "Vulnerabilities" i resoldre o justificar les marcades com a "Security Hotspots". La resta d'incidències marcades com a "Bug" o "Code Smell" s'hauran d'intentar minimitzar.

Per enviar el codi del projecte al sonarqube corporatiu simplement cal tenir instal·lat sonarscanner: https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner/ [9]

I executar la comanda des de l'arrel del projecte:
/c/_CARPETA_AL_SONNAR_SCANNER_/sonar-scanner/bin/sonar-scanner.bat
-D"sonar.host.url=http://su0244.corpo.ad.diba.es:9000/"
-D"sonar.projectKey=_CODI_DEL_PROJECTE_"
-D"sonar.login=_TOKEN_ACCESS_"
-D"sonar.exclusions=public/**,config/**,webpack.config.js"

On:
_CARPETA_AL_SONNAR_SCANNER_ = Carpeta on hagis descarregar el sonnar scaner
_CODI_DEL_PROJECTE_ = Codi del projecte, per exemple: REI, DELFOS; etc...
_TOKEN_ACCESS_ = Token d'accés al sonar scanner coporatiu
exclusions = carpetes i fitxers que no s'han d'analitzar

19. Seguretat

Tot nou projecte o evolutiu ha de superar el test bàsic de seguretat. Tant abans de planificar un evolutiu com després de realitzar-lo, cal executar aquest test per identificar les vulnerabilitats i actuacions relacionades amb la seguretat que cal resoldre, a més de les pròpies de l'evolutiu.

Aquest test bàsic consisteix principalment a verificar que no existeixen dependències amb vulnerabilitats conegudes i que les llibreries i frameworks utilitzats es mantenen actualitzats.

Cal tenir en compte que aquestes eines només detecten vulnerabilitats conegudes en dependències de tercers. No substitueixen una revisió de seguretat funcional o de codi, ja que no detecten problemes com ara vulnerabilitats XSS, CSRF, errors d'autenticació o autorització, exposició de secrets, configuracions insegures o altres riscos recollits a l'OWASP Top 10.

Stack	Comanda principal	Notes
npm (React, Vue i altres)	npm audit + npm outdated	Inclou npm audit fix per intentar corregir automàticament les vulnerabilitats solucionables
yarn	yarn audit	Recomanable complementar-lo amb la revisió de dependències desactualitzades
Angular	npm audit + npm outdated + ng update	ng update permet identificar paquets Angular i dependències associades pendents d'actualitzar
Symfony / PHP	composer audit	Comanda oficial de Composer (recomanada)
Symfony CLI	symfony check:security	Basat en la base de dades de seguretat de Symfony
Drupal	composer audit + composer outdated "drupal/*"	Drupal utilitza Composer internament

Cal executar totes les comandes rellevants segons el tipus de projecte. Per exemple:

- Un projecte Drupal haurà de superar:
  - drush pm:security
  - composer audit
- Un projecte Symfony amb components Javascript haurà de superar:
  - composer audit
  - symfony check:security
  - npm audit o yarn audit
  - npm outdated (si utilitza npm)
- Un projecte Angular haurà de superar:
  - npm audit o yarn audit
  - npm outdated (o equivalent)
  - ng update
- Un projecte React o Vue haurà de superar:
  - npm audit o yarn audit
  - npm outdated (o equivalent)

A més de les comandes anteriors, en projectes Javascript és recomanable revisar periòdicament les dependències

no utilitzades o abandonades i mantenir actualitzades les dependències principals del framework (Angular, React, Vue, Next.js, Nuxt, etc.).

L'aparició de vulnerabilitats classificades com a *High* o *Critical* ha de ser analitzada i resolta abans de la posada en producció, llevat que existeixi una justificació tècnica documentada i acceptada pel responsable del projecte.

20. Empreses

Les webs desenvolupades fora, que s'hauran d'allotjar en el nostra servidor, han de tenir en compte el següent:

- Els entregables per desplegar la web, hauran de venir amb un document d'instal·lació, amb els passos a a seguir per fer la correcte instal·lació.
- En els entregables les carpetes com vendor, var, web/core, web/modules/contrib, node_modules, etc... estaran buides, ja que les carregarem a través del composer. La carpeta de temporals i logs també hauria d'estar buida.
- Tots els complements instal·lats per fer proves o debbugging que no s'utilitzin s'han d'haver netejat abans de l'entrega.
- Cada entrega de l'empresa haurà d'estar al dia de les actualitzacions de seguretat, sense codi obsolet o no suportat.
- La web ha de ser accessible (https://www.diba.cat/accessibilitat) [10], els principals validadors utilitzats són WAVE o https://validator.w3.org [11].
- El codi entregat ha de passar pel sonarqube coporatiu i no tenir cap alerta de seguretat ni hotspot per resoldre.
- Es validarà que les llibreries i dependències del projecte es generen i carreguen via composer, yarn, webpack, etc... des dels repositoris oficials.
- Es validarà que el log d'aplicació estigui configurat com a rotatiu.
- Cal superar els checks de seguretat definits en l'apartat de seguretat

Categories: Plataforma PHP

URL d'origen: https://comunitatdstsc.diba.cat/wiki/resum-de-la-plataforma-php

Enllaços:
[1] https://mariadb.com/kb/en/mariadb-vs-mysql-compatibility/
[2] https://comunitatdstsc.diba.cat/wiki/desenvolupament-corporatiu-amb-drupal
[3] https://symfony.com/releases
[4] https://api-platform.com/
[5] https://maqueta.diba.cat/v2
[6] https://getcomposer.org/
[7] https://comunitatdstsc.diba.cat/wiki/desplegaments-php
[8] http://192.168.156.175
[9] https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner/
[10] https://www.diba.cat/accessibilitat)
[11] https://validator.w3.org