Diagrama

Descripción generada automáticamenteEncara que aquest patró és més freqüent en aplicacions web, s’ha utilitzat en el projecte perquè els components es divideixen en grups, fet que facilita la escalabilitat alhora de desenvolupar codi.

Per a administrar la base de dades *MySQL* s’ha utilitzat la eina *PhpMyAdmin* [13] la qual permet realitzar de manera completa les operacions bàsiques de la base de dades mitjançant la seva interface web intiutiva. Per la connexió de l’aplicació amb la base de dades s’ha realitzat tot en local a través d’un contenidor *Docker* [14]. *Docker* permet crear contenidors portables per a aplicacions software i executar-se en cualsavol màquina. Per la implementació dels contenidors s’ha utilitzat el propi software de *docker desktop* el qual ha permès instanciar els contenidors juntament amb la imatge oficical de *Phpmyadmin*, descarregada de la pàgina oficial de docker [15]. Tota la configuració s’ha realitzat a través del fitcher docler-compose.ymal on es defineixen els dos serveis: MySQL y PhpMyAdmin. Cada servei esta a dins d’un contenidor i es comuniquen entre ells per intercanviar l’informació. Els dos contenidors estan enllaçats entre sí ja que en el contenidor PhPMyadmin se li expecifica que depèn del servei MySQL y que aquest no es podrà incicar sense que prèviament hagi iniciat el contenidor MySQL. És tracta de que el contenidor MySQL carregui totes les dades de la base de dades i les comuniqui al contenidor PhPMyadmin perquè aquest únicament les mostri en la interfície web. Resulta necessari disposar dels dos contenidors ja que PhpMyAdmin es una eina d’administració de MySQL, per tant si només tinguessim el contenidor PhPMyadmin al no tenir el contenidor principal, dins de la interfice web de PhPMyAdmin, no tindríem les dades de la base de dades.

En el llenguatje java s’ha especificat el mètode ***EnviarCorreu()*** per poder enviar correus, dividit en:

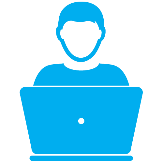
* **Session:** Representa la connexió amb el servidor gmail. En la seva configuració s’ha despecificar el seguent:
  + Usuari: [controltasques@gmail.com](mailto:controltasques@gmail.com)
  + Contrasenya: proporcionada per google
  + Protocol: SMTP i TLS (si esta dispoible)
  + Port: 587 (determinat)
* **Transport:** Envia el misatge cridant al métode **getTransport(“smpt”)** de la clase sessio.
* **Message:** S’especifica l’assumpte i el cos del missatge.

Estan enllaçats entre si, ja que en el contenidor PhPMyadmin s'especifica que depèn del servei MySQL i que aquest no es podrà iniciar sense que prèviament hagi iniciat el contenidor MySQL.

, per tant, si només tinguéssim el contenidor PhPMyadmin com que no tindríem del contenidor principal, dins de la interfície web de PhPMyAdmin, no tindríem les dades de la base de dades

Actualment, una de les possibles millores seria netejar el codi i fer-lo més eficient, ja que tot i haver seguit un patró de disseny, hi han vàries parts que es poden optimitzar.

He complert el primer objectiu controlant les tasques amb l'apartat de tasques de l'aplicació. Pel que fa al segon objectiu l'he aconseguit amb l'apartat elements i respecte al tercer objectiu s'ha complert amb l'apartat de grups. Aquests tres primers objectius eren els més prioritaris del projecte i s'han assolit.



Actualització

Petició

Resposta

Comunicació de dades

**Vista**

**Model**

**Controlador**