

# **Pràctica 3 DAT**

# Realització d'un Fòrum simple

*Per a una millor visualització del codi d'aquesta pràctica, podeu veure-la també a*

*[https://github.com/akaKush/DAT\\_UPC/tree/main/P3](https://github.com/akaKush/DAT_UPC/tree/main/P3)*

L'objectiu d'aquesta pràctica és fer una webapp clàssica (server-side app) utilitzant un framework específic de DAT per facilitar el desenvolupament.

Es treballen els següents aspectes:

- disseny basat en una arquitectura Model-View-Controller (MVC).
- reutilització de classes i patrons proporcionats per un framework (DatFw).
- manteniment de l'estat de les sessions amb els clients.
- autenticació d'usuaris i funcionalitat depenent de l'usuari.
- accés a bases de dades per al manteniment persistent de la informació.
- generació dinàmica del HTML. Definició dels components de visualització amb el llenguatge de plantilles de DatFw.

Crearem una webapp que permeti la creació de diferents temes de discussió (fòrums) i la creació de qüestions (tòpics) i respostes.

Abans però veurem un parell d'exemples:

# Hello World

Tracta d'una webapp que genera dinàmicament una pàgina HTML amb el text "Hello World". Es podria fer de forma estàtica editant el fitxer HTML, però així veiem una introducció a **DatFw**.

Primer creem un nou directori a dins del directori `practiques` (hello-world en el meu cas) i creem un fitxer a dins anomenat `hello.hs` i copiem el codi font:

```
{-# LANGUAGE OverloadedStrings #-}
{-# LANGUAGE TypeFamilies      #-}
{-# LANGUAGE QuasiQuotes       #-}

module Main
where
import Develop.DatFw
import Develop.DatFw.Template
import Network.Wai.Handler.CGI (run)

-- Definicions basiques del lloc

data HelloWorld = HelloWorld

instance WebApp HelloWorld

-- Rutes i 'dispatch'

instance RenderRoute HelloWorld where
    data Route HelloWorld = HomeR
    renderRoute HomeR    = ([], [])

instance Dispatch HelloWorld where
    dispatch =
        routing $ route ( onStatic [] ) HomeR [ onMethod "GET"
getHomeR ]

-- 'Handlers'

getHomeR :: HandlerFor HelloWorld Html
getHomeR = defaultLayout $ do
```

```

    setTitle "Hello"
    [widgetTempl | <h1>Hello World!</h1> |]

-- Inicialització3

main :: IO ()
main = -- CGI adapter: run :: Application -> IO ()
      toApp HelloWorld >=> run

```

Llavors editem en el mateix directori un fitxer `make-hello` amb el següent script de shell:

```

#!/bin/bash

main_file=hello.hs
exe_file=~/.public_html/practica3/hello.cgi

test -d ~/.public_html/practica3 || mkdir -p ~/.public_html/
practica3
source ~/.WEBprofe/usr/share/bin/do-make-exe.sh

```

Un cop ho tenim fet, li donem permisos d'execució i ja podem compilar/instal·lar l'exemple: (dins del directori (`practiques/hello-world`))

```

chmod +x make-hello
./make-hello

```

Si ens dóna algun error, seguir els passos que ens indica, en el meu cas he hagut de donar permisos de lectura/escriptura a la carpeta on està ubicat l'arxiu.

Llavors podem veure'l a [/public\\_html/practica3/hello.cgi](http://public_html/practica3/hello.cgi)

És important fixar-se amb l'enrutament d'aquest exemple, veiem que només tenim una sola ruta (`Route HelloWorld, HomeR`), amb la seva funció de renderització i una funció d'enviament de la petició al Handler corresponent.

# Hello World!

## To-Do List

En aquest segon exemple ja se'ns introdueix una aplicació amb una arquitectura més semblant a l'aplicació final. Aquesta fa ús de base de dades i diverses rutes de peticions.

Farem una llista de tasques que puguin estar en 2 estats: PENDENT o FET.

Primer necessitem baixar-nos i descomprimir l'arxiu del projecte al directori de pràctiques:

```
~/practiques$ curl http://soft0.upc.edu/dat/practica3/exemples/  
tasks.zip > tasks.zip  
~/practiques$ unzip tasks.zip  
~/practiques$ cd tasks  
~/practiques/tasks$ ls  
    bin build src
```

L'estat de la app es manté en una bbdd, la qual s'ha de crear amb les seves taules.

Utilitzem SQLite. Per crear-la executem `~/practiques/tasks$ bin/init-db`.

Per compilar i executar l'aplicació tenim el script `bin/make-tasks`, on hi podem veure:

```
#!/bin/bash

cd $(dirname $0)/..

# -- Configuration variables:
#     src_dirs
#     main_file
#     build_dir
#     cgi_dir
#     exe_file

src_dirs=src/haskell
main_file=src/haskell/Main.hs

build_dir=build

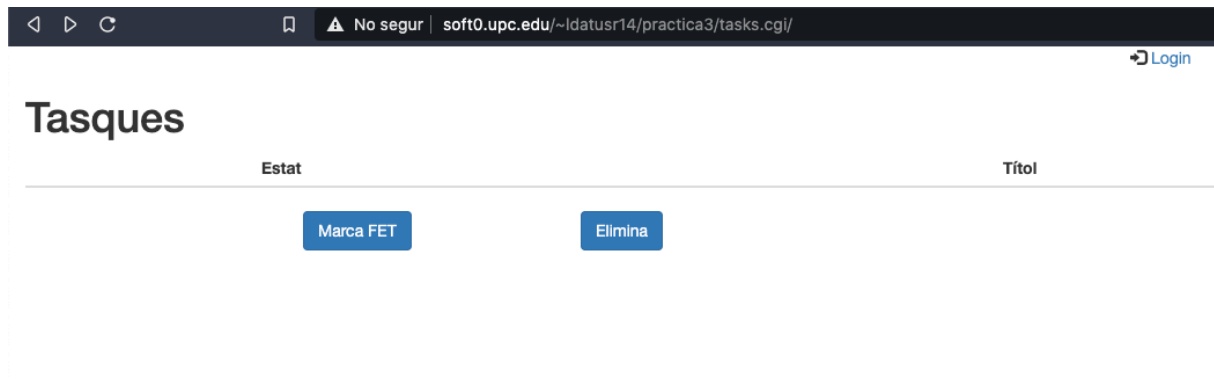
cgi_dir=~/.public_html/practica3
exe_file=$cgi_dir/tasks.cgi

test -d $cgi_dir || mkdir -p $cgi_dir
source ~WEBprofe/usr/share/bin/do-make-exe.sh
chmod 700 $exe_file
```

Per executar aquest arxiu fem `~/practiques/tasks$ bin/make-tasks` i veiem com s'instal·la el programa al directori de la practica3.

Si ens fixem en el fitxer `App.hs`, podem veure com en aquest tenim varies rutes de peticions:

```
instance Dispatch Tasks where
    dispatch = routing
        $ route ( onStatic [] ) HomeR
            [ onMethod "GET" getHomeR
              , onMethod "POST" postHomeR
            ]
    <|> routeSub (onStatic ["auth"]) AuthR getAuth
```



## Fòrum

Bé ara que ja hem vist com funcionen els exemples previs, passem a la pràctica.

Primer cal que descomprimim l'arxiu del projecte a la nostra zona d'usuari:

```
~/practiques$ curl http://soft0.upc.edu/dat/practica3/forums.zip
> forums.zip
~/practiques$ unzip forums.zip
~/practiques$ cd forums
~/practiques/forums$ ls
bin src
```

Per compilar la pràctica 3, com de costum tenim un script `bin/make-forums` que compila i instal·la el programa `forums.cgi` al directori `public_html/practica3`.

## Creació de la db

El SQL per crear la db del projecte es troba a `src/qlite/create-db.sql`. Aquest SQL té les taules necessàries del model (users, forums, topics i posts), i insereix les dades d'un fòrum per poder començar a fer proves.

Si entrem a l'arxiu `create-db.sql`, podem canviar els diversos camps que tenim, com els usuaris, els forums, etc. Jo els he deixat tal i com estan, ja que ja havia executat els següents passos:

```
~/practiques/forums$ mkdir ~/sqlite-dbs
~/practiques/forums$ chmod 700 ~/sqlite-dbs
~/practiques/forums$ sqlite3 ~/sqlite-dbs/forums.db < src/sqlite/
create-db.sql
```

Ens ha creat un directori `sqlite-dbs` amb el fitxer `forums.db` al nostre usuari.

*NOTA: Per poder accedir bé a la BD, hem de modificar el fitxer **App.hs** i afegir el nostre usuari a la ruta per trobar la BD: `forumsDbName = "/home/pract/LabDAT/ldatusr14/sqlite-dbs/forums.db"`*

## Codi font

Al directori `/practiques/forums/src/haskell` podem veure tots els fitxers que formen part del codi principal.

Utilitzarem el framework `DatFw`.

Se'ns demana acabar de completar els fitxers **Handler** i **View** principalment, ja que els altres estan pràcticament complets.

El fitxer `Handler` és el que conté les accions a executar quan es rep una petició.

## Handler.hs

```
getHomeR :: HandlerFor ForumsApp Html
getHomeR = do
    -- Get authenticated user
    mbuser <- maybeAuth
    -- Get a fresh form
    fformw <- generateAFormPost newForumForm
    -- Return HTML content
    defaultLayout $ homeView mbuser fformw
```



```

postHomeR :: HandlerFor ForumsApp Html
postHomeR = do
  user <- requireAuth
  (fformr, fformw) <- runAFormPost newForumForm
  case fformr of
    FormSuccess newtheme -> do
      now <- liftIO getCurrentTime
      runDbAction $ addForum newtheme now
      redirect HomeR
    _ ->
      defaultLayout $ homeView (Just user) fformw

getForumR :: ForumId -> HandlerFor ForumsApp Html
getForumR fid = do
  -- Get requested forum from data-base.
  -- Short-circuit (responds immediately) with a 'Not found'
  status if forum don't exist
  forum <- runDbAction (getForum fid) >>= maybe notFound pure
  mbuser <- maybeAuth
  -- Other processing (forms, ...)
  tformw <- generateAFormPost newTopicForm
  -- Return HTML content
  defaultLayout $ forumView mbuser (fid, forum) tformw

postForumR :: ForumId -> HandlerFor ForumsApp Html
postForumR fid = do
  user <- requireAuth
  forum <- runDbAction (getForum fid) >>= maybe notFound pure
  (tformr, tformw) <- runAFormPost newTopicForm
  case tformr of
    FormSuccess newtopic -> do
      now <- liftIO getCurrentTime
      runDbAction $ addTopic fid (fst user) newtopic now
      redirect (ForumR fid)
    _ ->
      defaultLayout (forumView (Just user) (fid, forum)
tformw)

getTopicR :: TopicId -> HandlerFor ForumsApp Html
getTopicR tid = do
  -- Get Topic

```

```

topic <- runDbAction (getTopic tid) >>= maybe notFound pure
-- Get authenticated user
mbuser <- maybeAuth
-- Get a fresh form
rformw <- generateAFormPost newReplyForm
-- Return HTML content
defaultLayout (topicView mbuser (tid, topic) rformw)

getDeleteTopicR :: TopicId -> HandlerFor ForumsApp Html
getDeleteTopicR tid = do
  user <- requireAuth
  topic <- runDbAction (getTopic tid) >>= maybe notFound pure
  runDbAction $ deleteTopic (tdForumId topic) tid
  redirect (ForumR (tdForumId topic))

getDeletePostR :: PostId -> HandlerFor ForumsApp Html
getDeletePostR pid = do
  user <- requireAuth
  post <- runDbAction (getPost pid) >>= maybe notFound pure
  topic <- runDbAction (getTopic (pdTopicId post)) >>= maybe
notFound pure
  runDbAction $ deletePost (tdForumId topic) (pdTopicId post)
pid
  redirect (TopicR (pdTopicId post))

postTopicR :: TopicId -> HandlerFor ForumsApp Html
postTopicR tid = do
  user <- requireAuth
  topic <- runDbAction (getTopic tid) >>= maybe notFound pure
  (rformr, rformw) <- runAFormPost newReplyForm
  case rformr of
    FormSuccess newreply -> do
      now <- liftIO getCurrentTime
      runDbAction $ addReply (tdForumId topic) tid (fst
user) newreply now
      redirect (TopicR tid)
    _ ->
      defaultLayout (topicView (Just user) (tid, topic)
rformw)

```

Veiem com hem afegit `getDeleteTopicR` `getDeletePostR` ja que així en cas de ser l'administrador o moderador podriem eliminar topics i posts.

## View.hs

Veiem les 3 vistes que tindrem al nostre cgi:

```
homeView :: Maybe (UserId, UserD) -> Widget ForumsApp -> Widget
ForumsApp
homeView mbuser fformw = do
    let isAdmin = maybe False (udIsAdmin . snd) mbuser
    forums <- runDbAction getForumList
    $(widgetTemplFile $ "src/templates/home.html")

forumView :: Maybe (UserId, UserD) -> (ForumId, ForumD) -> Widget
ForumsApp -> Widget ForumsApp
forumView mbuser (fid, forum) tformw = do
    let isMod = maybe False ((==) (fdModeratorId forum) . fst)
    mbuser
    topics <- runDbAction $ getTopicList fid
    $(widgetTemplFile $ "src/templates/forum.html")

topicView :: Maybe (UserId, UserD) -> (TopicId, TopicD) -> Widget
ForumsApp -> Widget ForumsApp
topicView mbuser (tid, topic) rformw = do
    forum <- runDbAction (getForum (tdForumId topic)) >>= maybe
    notFound pure
    let isMod = maybe False ((==) (fdModeratorId forum) . fst )
    mbuser
    replies <- runDbAction $ getPostList tid
    $(widgetTemplFile $ "src/templates/topic.html")
```

## Templates

Finalment veiem les interfícies que tindrem per visualitzar el cgi. Aquestes estan separades en 3 fitxers HTML, cadascun relacionat amb les funcionalitats vistes anteriorment.

Per a implementar bé les 3 vistes, m'he basat en el fitxer home.html que ja se'ns dona fet:

### home.html:

```
<h1>Marc Bosch - Pràctica 3 DAT - Creació d'un Fòrum</h2>

<table class="table table-striped table-condensed">
  <thead><tr><th>Categoria</th><th>Títol</th><th>Moderador</th><th>Creat</th><th>Topics</th><th>Posts</th><th>Última
activitat</th></tr></thead>
  <tbody>
    $forall{ (fid, forum) <- forums }
      <tr>
        <td>#{ fdCategory forum }</td><td><a href="@{ForumR
fid}">#{ fdTitle forum }</a></td>
        <td>^{ uidNameWidget (fdModeratorId forum) }</td>
        <td>^{ dateWidget (fdCreated forum) }</td>
        <td>#{ fdTopicCount forum }</td>
        <td>#{ fdPostCount forum }</td>
        <td>$maybe{ lastpid <- fdLastPostId forum }
^{pidPostedWidget lastpid} $end </td>
      </tr>
    $end
  </tbody>
</table>

$if{ isAdmin }
<h4>Afegeix un forum nou</h4>
<div class="row">
  <div class="col-sm-2"></div>
  <div class="col-sm-10">
    <form role="form" method="POST" action="@{HomeR}">
      ^{fformw}
      <button type="submit" class="btn btn-success">Nou fòrum</
button>
    </form>
  </div>
</div>
$end
```

## forum.html

```
<p><a href="@{HomeR}">Torna a la pàgina principal</a></p>

<h1>Llistat de Fòrums</h1>

<table class="table table-striped table-condensed">
  <thead><tr><th>Categoria</th><th>Títol</th><th>Moderador</th>
  <th>Creat</th><th>Topics</th><th>Posts</th></tr></thead>
  <tbody>
    <tr>
      <td>#{ fdCategory forum }</td><td>#{ fdTitle forum }</td>
      <td>^{ uidNameWidget (fdModeratorId forum) }</td>
      <td>^{ dateWidget (fdCreated forum) }</td>
      <td>#{ fdTopicCount forum }</td>
      <td>#{ fdPostCount forum } </td>
    </tr>
  </tbody>
</table>

<div class="bg-light">#{ fdDescription forum }</div>

<p>Topics:</p>
<table class="table table-striped table-condensed">
  <thead><tr><th>Qüestió</th><th>Per / Iniciada</th><th>Posts</th>
  <th>Última activitat</th></tr></thead>
  <tbody>
    $forall{ (tid, topic) <- topics }
    <tr>
      <td><a href="@{TopicR tid}"><strong>#{ tdSubject topic }
</strong></a></td>
      <td>^{ uidNameWidget (tdUserId topic) } / <span
class="small">^{ dateWidget (tdStarted topic) }</span></td>
      <td>#{ tdPostCount topic }</td>
      <td>$maybe{ lastpid <- tdLastPostId topic }
^{pidPostedWidget lastpid} $end </td>
      $if{ isMod }
      <td><a href="@{DeleteTopicR tid}">Remove</a></td>
      $end
    </tr>
  </tbody>
  $end
</table>
```

```

    </tbody>
</table>

$if{ isJust mbuser }
  <div class="row">
    <div class="col-sm-12">
      <form role="form" method="POST" action="@{ForumR fid}">
        ^{tformw}
        <button type="submit" class="btn btn-success">Nou Tòpic</
button>
      </form>
    </div>
  </div>
$end

```

## topic.html

```

<h1>Topics</h2>
<p><i>^{ uidNameWidget (tdUserId topic) } - ^{ dateWidget
(tdStarted topic)}</i></p>
<h4><strong>#{ tdSubject topic}</strong></h4>
<br>
<h4>Respostes:</h4>
<table class="table table-striped table-condensed">
  <thead><tr><th>User</th><th>Message</th><th>Date</
th>$if{ isMod }<th>Remove</th>$end</tr></thead>
  <tbody>
    $forall{ (rid, reply) <- replies }
      <tr>
        <td><strong>^{ uidNameWidget (pdUserId reply) }</
strong></td>
        <td>#{ pdMessage reply }</td>
        <td><span class="small">^{ dateWidget (pdPosted reply) }
</span></td>
        $if{ isMod }
        <td><a href="@{DeletePostR rid}">Elimina</a></td>
      $end
    </tr>
  $end
</tbody>
</table>

```

```

$if{ isJust mbuser } <div class="row">
  <div class="col-sm-12">
    <form role="form" method="POST" action="@{TopicR tid}">
      ^{rformw}
      <button type="submit" class="btn btn-success">Nova
Resposta al Tòpic</button>
    </form>
  </div>
</div>
$end

```

*Nota final: Per a poder tenir les funcionalitats d'eliminar tòpics i posts, he hagut de modificar també el fitxer **Found.hs** afegint les dues rutes necessaries. Adjunto el codi modificat del fitxer Found.hs a sota:*

```

instance RenderRoute ForumsApp where
  data Route ForumsApp =
    HomeR | ForumR ForumId | TopicR TopicId
    | DeletePostR PostId | DeleteTopicR TopicId |
AuthR (Route Auth)

  renderRoute HomeR    = ([], [])
  renderRoute (ForumR tid) = (["forums",toPathPiece tid], [])
  renderRoute (TopicR qid) = (["topics",toPathPiece qid], [])
  renderRoute (DeletePostR dpid) = (["delpost",toPathPiece
dpid], [])
  renderRoute (DeleteTopicR dtid) = (["deltopic",toPathPiece
dtid], [])
  renderRoute (AuthR authr) = let (path,qs) = renderRoute authr
in ("auth":path, qs)

```

# Resultat Final

El podeu veure mitjançant el següent link: <http://soft0.upc.edu/~ldatusr14/practica3/forums.cgi/>

Adjunto algunes fotos de les funcionalitats:

Usuari: **admin** [Tanca sessió](#)

## Marc Bosch - Pràctica 3 DAT - Creació d'un Fòrum

Categoria	Títol	Moderador	Creat	Topics	Posts	Última activitat
	<a href="#">Fòrum de prova</a>	admin	1 Nov 2020, 13:00	1	2	1 Nov 2020, 13:00
	<a href="#">Nou fòrum fet per Marc Bosch</a>	usuari1	17 May 2021, 20:21	2	2	17 May 2021, 20:22

**Afegeix un forum nou**

Titl:

Descripció:

Nom del moderador:

[Nou fòrum](#)



Usuari: **admin** [Tanca sessió](#)

[Torna a la pàgina principal](#)

## Llistat de Fòrums

Categoria	Títol	Moderador	Creat	Topics	Posts
	Nou fòrum fet per Marc Bosch	usuari1	17 May 2021, 20:21	2	2

Aquest és un fòrum de prova per comprovar la funcionalitat del cgi

Topics:

Qüestió	Per / Iniciada	Posts	Última activitat
<a href="#">TÒPIC DE PROVA fet per MARC BOSCH</a>	admin / 17 May 2021, 20:21	1	17 May 2021, 20:21
<a href="#">TÒPIC 2 AIXÒ FUNCIONA BÉ</a>	admin / 17 May 2021, 20:22	1	17 May 2021, 20:22

Title:

Description:

[Nou Tòpic](#)

La següent foto fixeuvos en que l'usuari és el **usuari1**, i també té la funcionalitat d'eliminar el post, dins el tòpic:

Usuari: **usuari1** [Tanca sessió](#)

## Topics

admin - 17 May 2021, 20:21

### TÒPIC DE PROVA fet per MARC BOSCH

Respostes:

User	Message	Date	Remove
admin	Aquest és el tòpic per comprovar que funciona bé la funcionalitat	17 May 2021, 20:21	<a href="#">Elimina</a>

Reply:

[Nova Resposta al Tòpic](#)

En canvi si accedim amb l'**usuari2**, el qual NO és moderador, veiem com no ens deixa eliminar res:

soft0.upc.edu/~ldatusr14/practica3/forums.cgi/topics/2

Usuari: **usuari2** [Tanca sessió](#)

## Topics

admin - 17 May 2021, 20:21

### TÒPIC DE PROVA fet per MARC BOSCH

Respostes:

User	Message	Date
admin	Aquest és el tòpic per comprovar que funciona bé la funcionalitat	17 May 2021, 20:21
usuari1	resposta al tòpic de prova creada per usuari1	17 May 2021, 20:57

Reply:

Introduce reply

[Nova Resposta al Tòpic](#)

Així com tampoc ens deixa crear cap més tòpic (amb l'usuari1 tampoc podem crear cap tòpic, només posts dins d'aquests):

soft0.upc.edu/~ldatusr14/practica3/forums.cgi/

Usuari: **usuari2** [Tanca sessió](#)

## Marc Bosch - Pràctica 3 DAT - Creació d'un Fòrum

Categoria	Títol	Moderador	Creat	Topics	Posts	Última activitat
	<a href="#">Fòrum de prova</a>	admin	1 Nov 2020, 13:00	1	2	1 Nov 2020, 13:00
	<a href="#">Nou fòrum fet per Marc Bosch</a>	usuari1	17 May 2021, 20:21	2	3	17 May 2021, 20:57