

Entrega Final

DAVICO - DURÁN - HERNÁNDEZ - MATURANA - TORT

GRUPO 14



Introducción

1 - Banco

2 - Exchange

Elementos relevantes

03

01

DIAGRAMA

Diagrama de arquitectura

02

TASK

Modularización

03

API

Separación authorize y authenticate

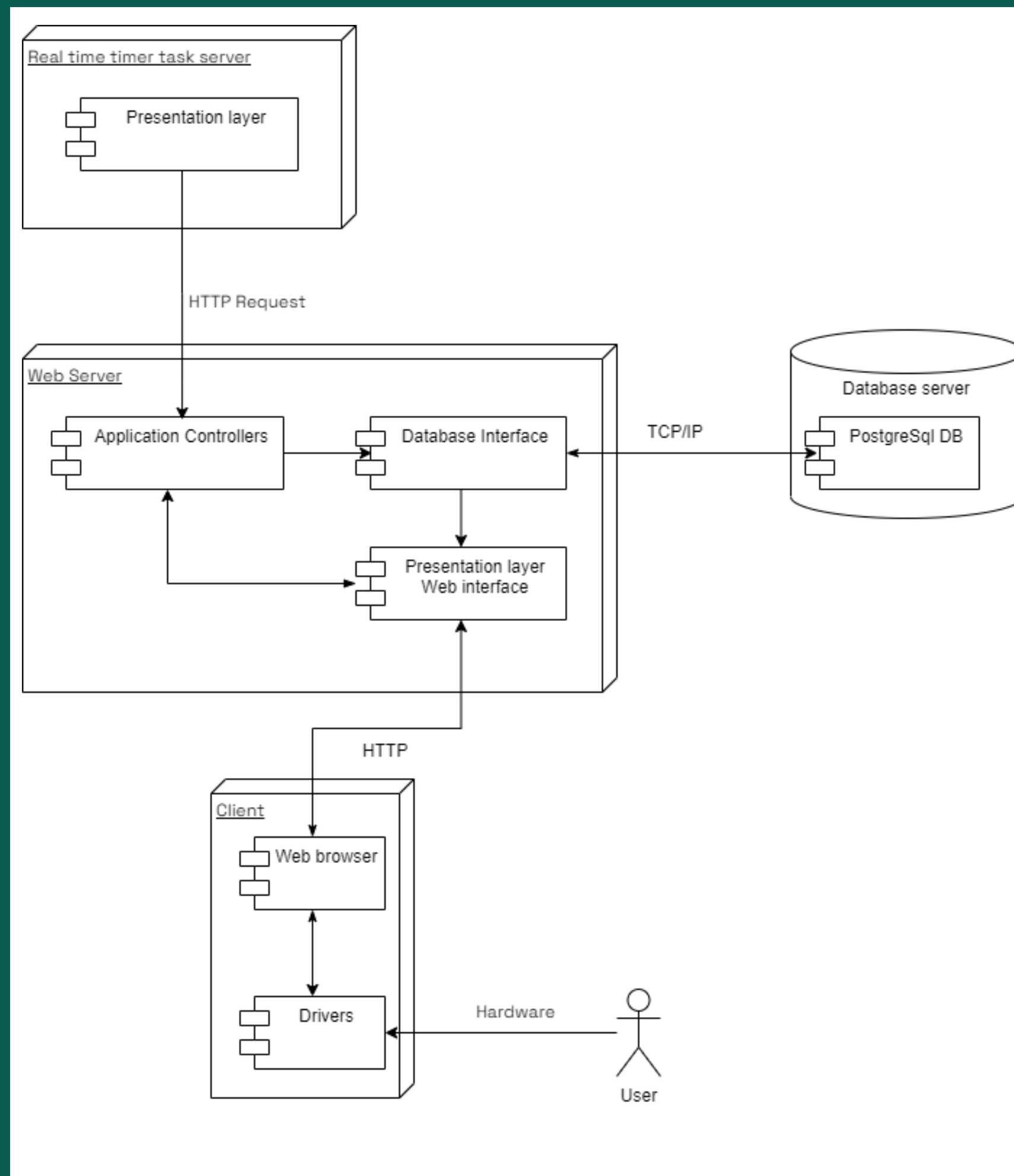


Diagrama de Arquitectura

Modularización de Task

```
JS index.js  X
routes > JS index.js > ...

You, a few seconds ago | 1 author (You)
1  const express = require('express');
2  const router = express.Router();
3  const axios = require('axios');
4  var cron = require('node-cron');
5
6  /* GET home page. */
7  router.get('/', (req, res) => {
8    res.render('index', { title: 'BankTask' });
9  });
10
11
12  cron.schedule('* * * * *', async () => {
13    console.log('running a task every hour');
14    axios
15    .post('http://bankdiseno.cl/update')
16    .catch(error => {
17      console.error(error)
18    });
19  });
20
21
22  module.exports = router;
23
```

Módulo Commands (API)

```
authenticate_user.rb • authorize_api_request.rb •
class AuthenticateUser
  #prepend SimpleCommand

  def initialize(email, password)
    @email = email
    @password = password
  end

  def encode_token
    if user
      JsonWebToken.encode(user_id: user.id)
    end
  end

  private

  attr_accessor :email, :password

  def user
    user = User.find_by_email(email)
    if user && user.valid_password?(password)
      return user
    end
  end
end
```

▼ commands

/* authenticate_user.rb

/* authorize_api_request.rb

```
authenticate_user.rb • authorize_api_request.rb •
class AuthorizeApiRequest
  prepend SimpleCommand

  def initialize(headers = {})
    @headers = headers
  end

  def check_token
    user
  end

  private

  attr_reader :headers

  def user
    if decoded_auth_token
      @user = User.find(decoded_auth_token[:user_id])
    end
  end

  def decoded_auth_token
    @decoded_auth_token ||= JsonWebToken.decode(http_auth_header)
  end

  def http_auth_header
    if headers['Authorization'].present?
      return headers['Authorization'].split(' ').last
    end
  end
end
```

Problemas encontrados

Problema 1

08

**Escasos
mensajes de
confirmación**



Problema 2

09

master

exchange-two / app / views / devise / registrations / new.html.erb

Avances en css

2 contributors

132 lines (93 sloc) | 3.87 KB

```
1 <%= form_for(resource, as: resource_name, url: registration_path(resource_name)) do |f| %>
2 .....
3 <div class="field">
4   <%= f.hidden_field :clp, :value => rand(10000..100000) %>
5 </div>
6 .....
7
8 <body class="">
9   <div class="wrapper ">
10
11
12   <div class="main-panel">
13
```



Inspeccionar
elemento

PRESENTACIÓN FINAL

Problema 2

20 lines (17 sloc) | 721 Bytes

```
1  class RegistrationsController < Devise::RegistrationsController
2    before_action :configure_sign_up_params, only: [:create]
3    before_action :configure_account_update_params, only: [:update]
4
5    protected
6
7    def after_update_path_for(resource)
8      edit_user_registration_path(resource)
9    end
10
11    def configure_sign_up_params
12      .....
13      devise_parameter_sanitizer.permit(:sign_up,
14        ..... keys: [:nombre, :apellido,
15        ..... :rut, :clp]) .....
16    end
17    def configure_account_update_params
18      devise_parameter_sanitizer.permit(:account_update, keys: [:nombre, :apellido,
19        ..... :rut])
20    end
21  end
```



**Agregar
saldo desde
el browser**

¿RUT?



Problema 3

```
92
93   create_table "users", force: :cascade do |t|
94     t.string "email", default: "", null: false
95     t.string "encrypted_password", default: "", null: false
96     t.string "reset_password_token"
97     t.datetime "reset_password_sent_at"
98     t.datetime "remember_created_at"
99     t.datetime "created_at", precision: 6, null: false
100    t.datetime "updated_at", precision: 6, null: false
101    t.string "first_name"
102    t.string "last_name"
103    t.index ["email"], name: "index_users_on_email", unique: true
104    t.index ["reset_password_token"], name: "index_users_on_reset_password_token", unique: true
105  end
106
```

Problema 4

RAMAS ACTIVAS



dev	Updated 20 minutes ago by frduran	1 16	#46	Merged	
readme-info	Updated 23 minutes ago by lldavico	1 15	#45	Merged	
FixAPIPrivacy	Updated 5 hours ago by nmaturationa97	1 12	#44	Merged	
random-price	Updated 2 days ago by jatort	1 7	#42	Merged	
envariables_prod	Updated last month by mistico0121	16 3		New pull request	
development	Updated last month by mistico0121	11 1		New pull request	
qolimprovements	Updated last month by ClaudioPrieto	29 0	#39	Merged	
ArregloEditorUsuario	Updated last month by aigou42	26 0	#38	Merged	
seeds	Updated last month by juanpad111	47 0	#36	Merged	
home_page	Updated last month by mistico0121	47 4		New pull request	
linegraphworking	Updated last month by mistico0121	88 0	#35	Merged	
linegraph2	Updated last month by perezpefaur	40 0	#33	Merged	
doc-diagram	Updated last month by ClaudioPrieto	106 0	#34	Merged	
linegraph	Updated last month by perezpefaur	66 0	#30	Merged	
remove-balance	Updated last month by ClaudioPrieto	106 0	#29	Merged	
quick-fix	Updated last month by ClaudioPrieto	107 0	#28	Merged	
bstemplate	Updated last month by mistico0121	101 4		New pull request	
api-doc	Updated last month by ClaudioPrieto	112 0	#27	Merged	
dockerize	Updated last month by mistico0121	112 0	#25	Merged	
partner_user	Updated last month by kenzoak	120 0	#22	Merged	

Problema 5

El peor método



```
22 def show
23   @current_version = @currency
24   @previous_version = @currency.paper_trail.previous_version
25   @versiones = []
26   @versiones << @current_version
27   if @previous_version != nil
28
29     print("")
30   end
31
32   # Obtiene a lo más las 30 ultimas versiones de la moneda
33   for i in (0..29) do
34     if @previous_version != nil
35       @versiones << @previous_version
36       @current_version = @previous_version
37       @previous_version = @current_version.paper_trail.previous_version
38     end
39   end
40
41   # Para utilizar Chartkick, se le debe pasar un objeto de tipo {[key0:value0], [key1:value1]} y así
42   # Y como realizo la query "a mano" en el controlador, debo crearlo acá el objeto
43   version_j = @versiones.each.map{|v| {sellvalue: v.sellvalue, updated_at: v.updated_at} }
44   version_j = version_j.collect{|i| [i[:updated_at], i[:sellvalue]]}
45
46   # Porcentajes
47   @previous_version2 = @currency.paper_trail.previous_version
48   if @previous_version2 != nil
49     @current_ver_value_percent = (100*(@currency.sellvalue - @previous_version2.sellvalue)/@previous_version2.sellvalue ).round(2)
50
51   else
52     @current_ver_value_percent = 0
53   end
54
55   @v_json = version_j
56
57 end
```

Code Smell: Bloaters - Long Method

Conclusiones

- Code smells muy comunes, manifiestan mayor complejidad para entender el código y alertan sobre posibles errores y/o bugs en el futuro.
- Desacoplamiento facilita mantenimiento de código y escalabilidad del software
- Importancia de una buena documentación
- Importancia de que el código sea autoexplicativo para nuevos desarrolladores.



Muchas Gracias

DAVICO - DURÁN - HERNÁNDEZ - MATURANA - TORT

GRUPO 14