

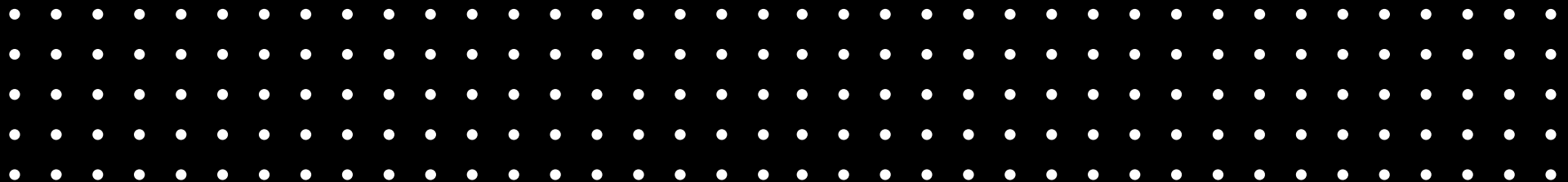
A decorative graphic consisting of a grid of small black dots arranged in 10 rows and 20 columns, located in the top-left corner of the slide.

REFLEXIONES DEL PROYECTO

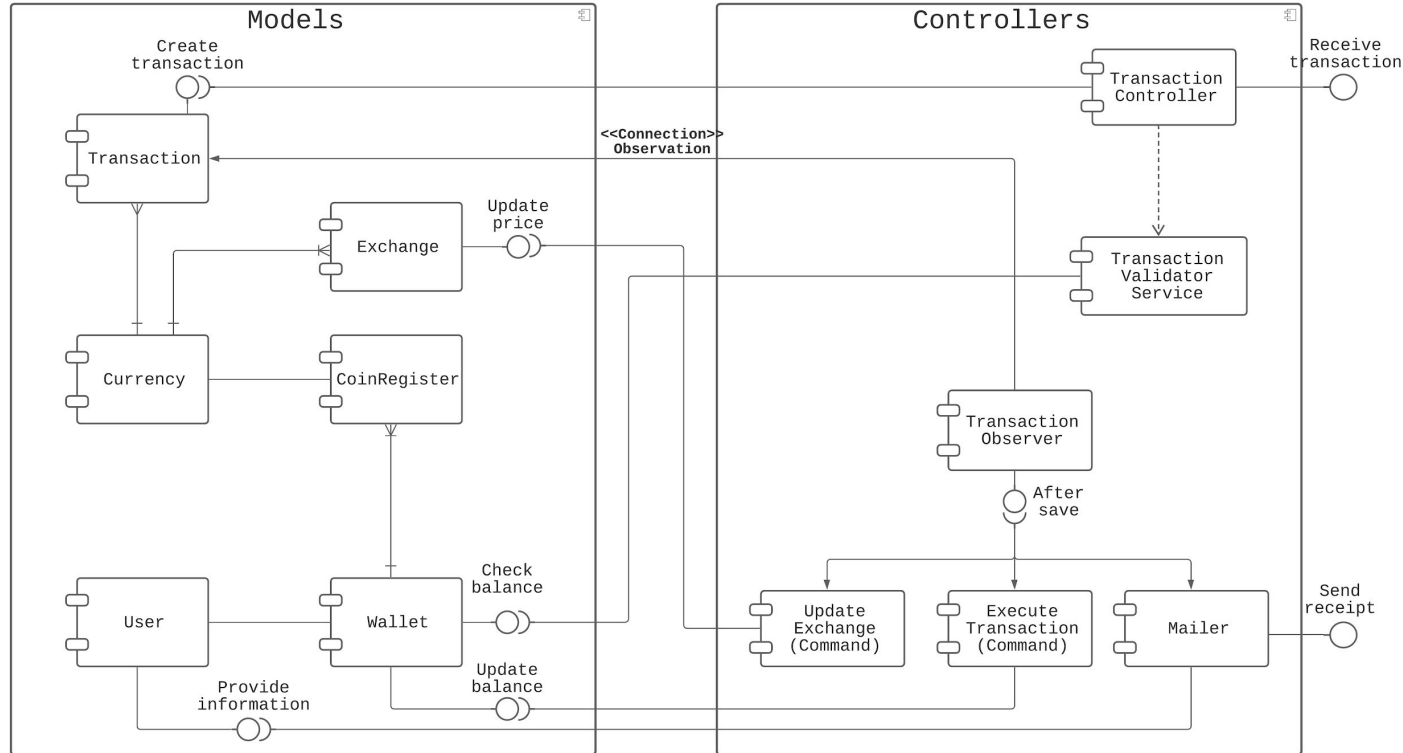
Grupo 5

01

DECISIONES RELEVANTES DE DISEÑO



PRINCIPIOS FUNDAMENTALES Y DIAGRAMA



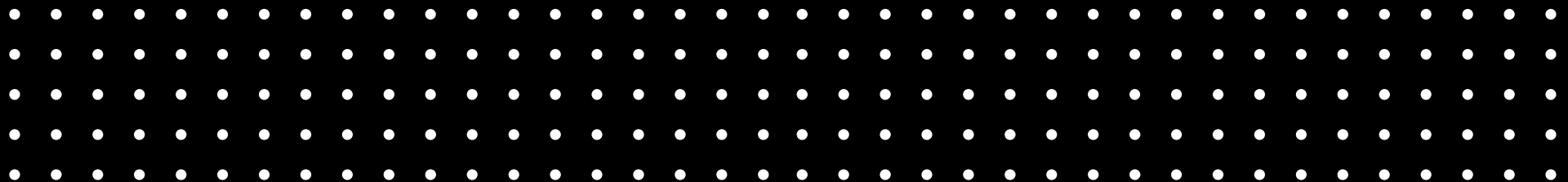
PATRÓN DE DISEÑO

```
1  class TransactionObserver < PowerTypes::Observer
2    after_save do
3      object
4
5      ExecuteTransaction.for(transaction: object)
6
7      UpdateExchangePrice.for(transaction: object)
8
9      wallet = Wallet.find object.wallet_id
10     user = User.find wallet.user_id
11     TransactionMailer.with(user: user, transaction: object).notification_mail.deliver_now
12   end
13 end
14
```

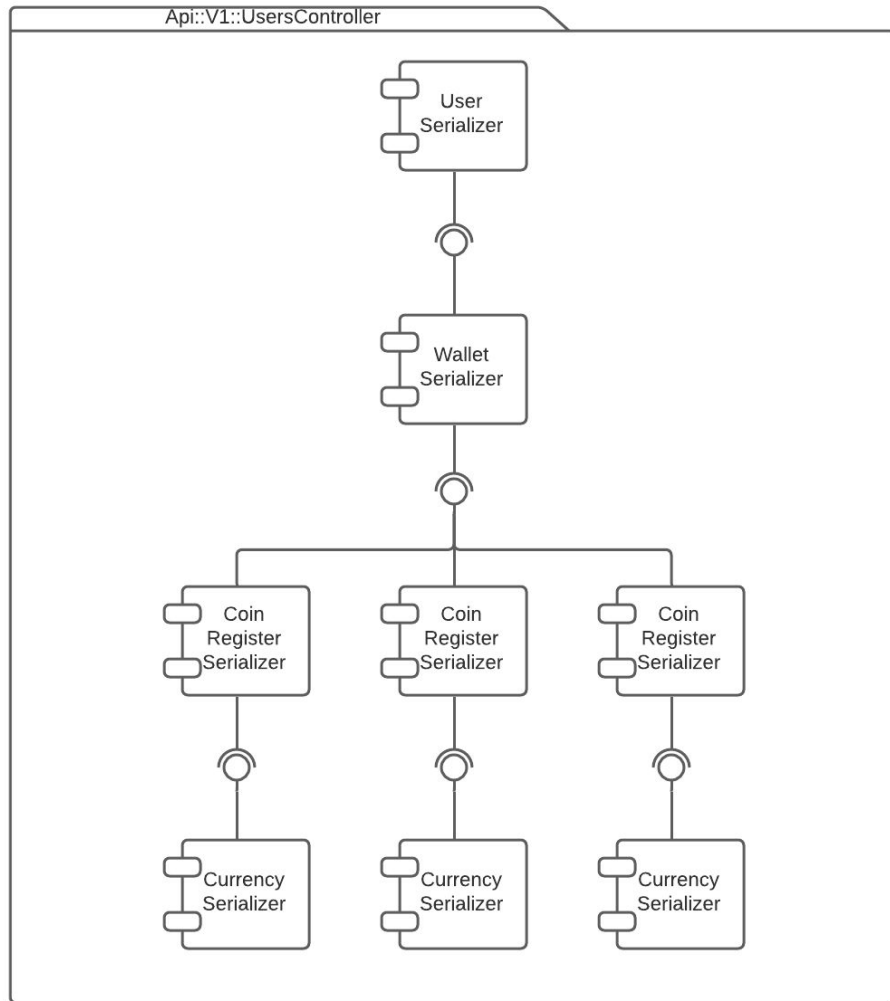


02

OTROS ASPECTOS RELEVANTES

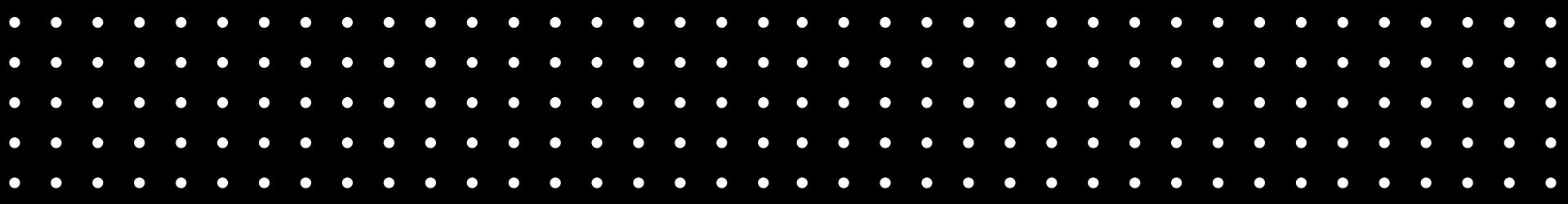
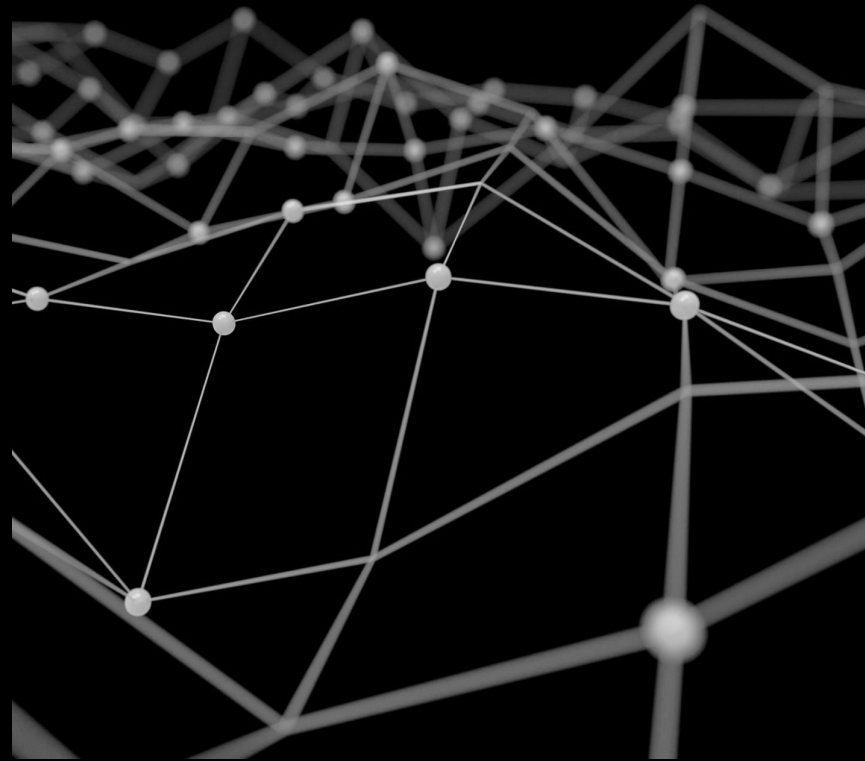


API



03

PROBLEMAS Y BUENAS SOLUCIONES



REFACTORING

Big linting incoming #32

 Merged yerkko merged 1 commit into `dev` from `bugfix/linting`  on 21 Oct



Conversation 1



Commits 1



Checks 0



Files changed 63



szabala commented on 21 Oct



Police came to my house, had to hide code under carpet.
please review! 🙏



Big linting incoming ...

be281e1

Rubocop, brace yourself



szabala requested a review from yerkko on 21 Oct



yerkko commented on 21 Oct



Nice pull request mate. Do they call you *The Linteringer* ?

CODE SMELLS & REFACTORING

```
1 class UpdateExchangePrice < PowerTypes::Command.new(:transaction)
2   def perform
3
4     btbf_price_changer = 0.03
5     bth_price_changer = 0.08
6
7     if @transaction.bought.initials != "CLP"
8
9       exchange = Exchange.find_by(buying: @transaction.bought)
10
11       case exchange.selling.initials
12       when "BTF"
13         exchange.price *= (1 - btbf_price_changer)
14       when "BTH"
15         exchange.price *= (1 - bth_price_changer)
16       end
17
18     else
19
20       exchange = Exchange.find_by(selling: @transaction.sold)
21
22       case exchange.buying.initials
23       when "BTF"
24         exchange.price *= (1 + btbf_price_changer)
25       when "BTH"
26         exchange.price *= (1 + bth_price_changer)
27       end
28       exchange.save!
29     end
30   end
31 end
```



```
1 class UpdateExchangePrice < PowerTypes::Command.new(:transaction)
2   def perform
3     bought_currency = Currency.find(@transaction.bought_id)
4     exchange = Exchange.find_by(buying: @transaction.bought_id, \
5       selling: @transaction.sold_id)
6
7     if bought_currency.initials != "CLP"
8       # Comprar crypto
9       currency = Currency.find(exchange.buying_id)
10
11       new_price = exchange.price * (1 + currency.price_changer)
12     else
13       # Vender crypto
14       currency = Currency.find(exchange.selling_id)
15
16       new_price = exchange.price * (1 - currency.price_changer)
17     end
18     exchange.update(price: new_price)
19   end
20 end
```

Switch Statements
Duplicated Code

PRINCIPIOS SOLID & REFACTORIZING

```
1 #POST /transactions
2 def create
3   sold = Currency.where(id : params[:sold])
4   bought = Currency.where(id: params[:bought])
5   if validate_transaction
6     @transaction = Transaction.new(date: Time.now, wallet_id: current_user.wallet.id, \
7       sold_qty: params[:sold_qty], bought_qty[:bought_qty], sold: sold, bought: bought)
8
9   respond_to do |format|
10     if transaction.save
11       format.html do
12         redirect_to @transaction, notice: 'Transaction was succesfully created.'
13       end
14       format.json { render :show, status: :created, location: @transaction }
15     else
16       format.html {render :new}
17       format.json do
18         render json: @transaction.errors, status: :unprocessable_entity
19       end
20     end
21   end
22 end
```

```
1 #POST /transactions
2 def create
3   sold = Currency.find(transaction_params[:sold])
4   bought = Currency.find(transaction_params[:bought])
5   sold_qty = transaction_params[:sold_qty]
6   bought_qty = CalculateExchange.for(sold_currency: sold, \
7     bought_currency: bought, qty: sold_qty)
8
9   validation_service = TransactionValidatorService.new(wallet: @wallet)
10  @transaction = Transaction.new(date: Time.zone.now, wallet_id: @wallet.id, \
11    sold_qty: sold_qty, bought_qty: bought_qty, sold: sold, bought: bought)
12
13  respond_to do |format|
14    if validation_service.validate(@transaction) && @transaction.save
15      render.successful(format)
16    else
17      render_error(format)
18    end
19  end
20 end
```

Single-responsability
principle

DDD

Libro de Órdenes

1

Precio(NGN)	Cantidad(BTC)	Total
9574403	0.214932	2,057,845.586
9570000	0.038643	369,813.510
9569178	0.000060	574.151
9568000	2.138640	20,462,507.520
9567225	0.126000	1,205,470.350
9564538	0.006754	64,598.890
9564529	0.038192	365,288.492
9564528	0.020145	192,677.417
9550305	0.003498	33,406.967

COMPRAR

VENDER

Limit

Market

Stop-Limit

- NGN

Precio

9532465

NGN

Cantidad

BTC

Inicia sesión

Registrarse

Realizar Orden

COMPRAR

VENDER

Limit

Market

Stop-Limit

- BTC

Precio

9532465

NGN

Cantidad

BTC

Inicia sesión

Registrarse

Activos

Compra con EUR

Depositar

Retirar

VS

Exchanges


BUY 1.0 Badtherium BTH for 52.907 Peso Chileno CLP

BUY 1.0 Peso Chileno CLP for 0.018 Badtherium BTH

BUY 1.0 BitFake BTF for 688.463 Peso Chileno CLP

BUY 1.0 Peso Chileno CLP for 0.001 BitFake BTF

Fix routes bug and after sign up redirect #35

 Merged ifieldhouse merged 1 commit into `dev` from `bugfix/sign_up` on 29 Oct

 Conversation 2

 Commits 1

 Checks 0

 Files changed 7



szabala commented on 28 Oct



Had a bug plague. Called the terminator.



Fix routes bug and after sign up redirect

1a13d1c



szabala requested a review from ifieldhouse on 28 Oct



yerkko commented on 28 Oct



@ifieldhouse please review this. You're the reviewer who reviews.



ifieldhouse approved these changes on 29 Oct

[View changes](#)

ifieldhouse left a comment



Haha! Good!

NO TESTING

**Muchas
gracias!**

