Bank twelve - Exchange Four

Grupo 24

Felipe Cabrera - Martín Cadagan - Camila Chávez - Mariana Ortega - Joaquín Ossandón

Índice

Flujo más relevante

Implementación Cronjob

Calidad de código

Refactoring

Conclusiones

Flujo de inversiones


Bank Twelve
Mis Cuentas
Transferir
Invertir
Retiro Inversiones
Fondo de Inversión
Cerrar Sesión
¡Bienvenido(a) a Bank Twelve!
Nos preocupamos por tu futuro
Revisar Mis Cuentas
Flujo de Inversiones

Bank Twelve Mis Cuentas Transferir Invertir Retiro Inversiones Fondo de Inversión Cerrar Sesión

Nueva Inversión

Tu saldo disponible es $79491

Monto a invertir

20000

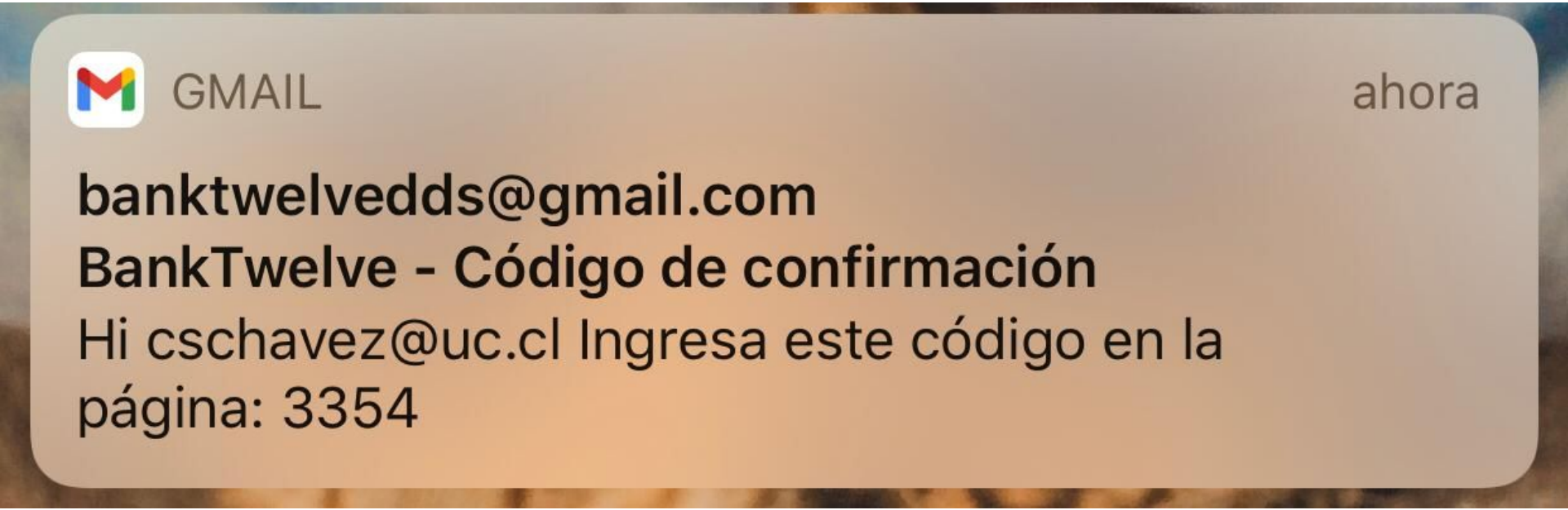
Invertir

Confirmar la inversión

Revisa tu mail e ingresa el código que te enviamos para confirmar la inversión

Código

Confirmar


GMAIL
ahora
banktwelvedds@gmail.com
BankTwelve - Código de confirmación
Hi cschavez@uc.cl Ingresa este código en la página: 3354

Inversión Exitosa

Monto Invertido 20000

¡Tu dinero está en las mejores manos!

Retiro de Fondos de Inversión

Tienes $20000 en tu cuenta de ahorro

Monto a retirar

Retiro

Retiro de Fondos de Inversión

Tienes $19719 en tu cuenta de ahorro

Monto a retirar

Retiro

Flujo más relevante

Implementación Cronjob

Calidad de código

Refactoring

Conclusiones

Implementación CronJob

1

Agregar gemas Sidekiq y Sidekiq-scheduler

Implementación CronJob

1

Agregar gemas Sidekiq y Sidekiq-scheduler

2

Añadir un volumen de Redis

9 ■■■■■ docker-compose.yml

```
@@ -10,5 +10,14 @@ services:
10  10          POSTGRES_PASSWORD: ''
11  11        volumes:
12  12        - postgresql_data:/var/lib/postgresql/data
    13  +   redis:
    14  +     container_name: 'redis2'
    15  +     image: 'redis:alpine'
    16  +     ports:
    17  +       - '127.0.0.1:6379:6379'
    18  +     volumes:
    19  +       - 'redisdata:/data'
    20  +
13  21    volumes:
14  22      postgresql_data:
    23  +   redisdata:
```

Implementación CronJob

1. Agregar gemas Sidekiq y Sidekiq-scheduler
2. Añadir un volumen de Redis
3. Crear el Job

obs > investment_job.rb

```
...
class InvestmentJob < ApplicationJob
    queue_as :default

    def perform
        all_accounts = Account.all
        all_accounts.each do |account|
            random_number = rand -0.03..0.03
            account.saving_balance *= (1 + random_number)
            account.save
        end
    end
end
```

Implementación CronJob

1. Agregar gemas Sidekiq y Sidekiq-scheduler
2. Añadir un volumen de Redis
3. Crear el Job
4. Configurar el archivo sidekiq.yml

sidekiq.yml > ...

You, seconds ago | 2 authors (You and

```yaml
production:
  :verbose: false
  :concurrency: 5
  :timeout: 30
:queues:
  - default
:schedule:
  InvestmentJob:
    cron: '0 * * * *'
    queue: default
    enabled: true
```

Implementación CronJob

1. Agregar gemas Sidekiq y Sidekiq-scheduler
2. Añadir un volumen de Redis
3. Crear el Job
4. Configurar el archivo sidekiq.yml
5. Procfile.dev

Flujo más relevante

Implementación Cronjob

Calidad de código

Refactoring

Conclusiones

Calidad del código

- Mejora del Cyclomatic Complexity
- Métodos más cortos
- Codalyze

```
3   -   class BuyController < ApplicationController
4   -     protect_from_forgery unless: -> { request.format.json? || request.format.xml? }
5   -     def create
6   -
7   -       if params[:token] and params[:amount]
8   -         if User.where(authentication_token: params[:token]).exists?
9   -           user = User.find_by(authentication_token: params[:token])
10  -           amount = params[:amount]
11  -
12  -           @transaction = Transaction.new()
13  -           @transaction.user_id = user.id
14  -           @transaction.tipo = "compra BTF"
15  -           @transaction.dateCreated = Time.new
16  -           @transaction.BTF_amount = amount
17  -
18  -           @uniqueBitfakeCurrency = Bitfake.find(1)
19  -           @wastedCLP = (amount.to_f) * (@uniqueBitfakeCurrency.purchaseprice)
20  -           @transaction.amount = @wastedCLP
21  -
22  -           if user.CLP_amount - @wastedCLP >= 0
```

Calidad del código

Code Complexity Report

> So if you want to go fast, if you want to get done quickly, if you want your code to be easy to write, make it easy to read.
>
> — Clean Code: A Handbook of Agile Software Craftsmanship *Robert C. Martin*

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
create	3	61	7	48	0

Calidad del código

Code Complexity Report

So if you want to go fast, if you want to get done quickly, if you want your code to be easy to write, make it easy to read.

— Clean Code: A Handbook of Agile Software Craftsmanship *Robert C. Martin*

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
create	3	13	4	11	0
aux_create	15	31	3	17	0
aux_generate_transaction	33	43	1	11	2
aux_transaction_save	45	53	1	9	4

Flujo más relevante

Implementación Cronjob

Calidad de código

Refactoring

Conclusiones

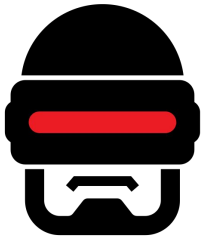
Refactoring

Para mejorar la calidad del código se realizaron dos acciones:

1. Refactorización en base a Rubocop 

2. Agregar comentarios explicativos al código por función **# Explanation comment**

Rubocop



RuboCop

```
74 files inspected, 257 offenses detected
```

refactoring the code with rubocop, all fixed 🔥

master (#14)

fncabrera committed 2 days ago

Showing 38 changed files with 351 additions and 331 deletions.

```
72 files inspected, no offenses detected
```

Ejemplo cambios

BuyController > Método “create”:

- Single-responsibility principle
- Bloaters - Long Method
- Variables innecesarias

buy_controller.rb

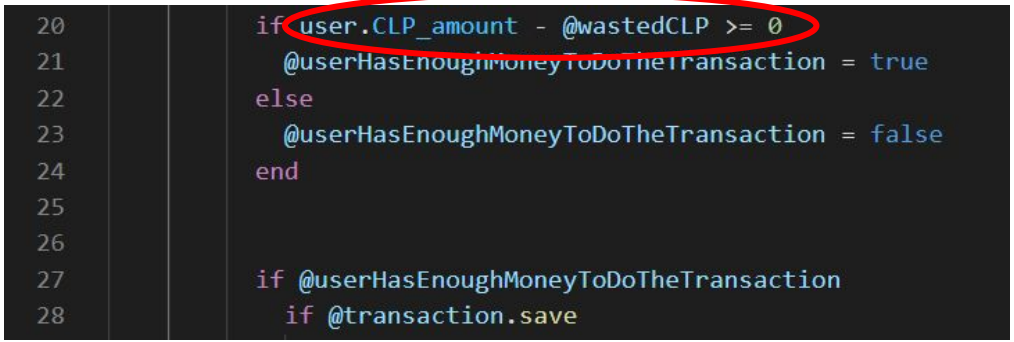
```
1   class BuyController < ApplicationController
2       protect_from_forgery unless: -> { request.format.json? || request.format.xml? }
3       def create
4
5         if params[:token] and params[:amount]
6           if User.where(authentication_token: params[:token]).exists?···
49          else
50            render json: { error: 'Access denied' }.to_json, status: :unauthorized
51          end
52        else
```

```
20            if user.CLP_amount - @wastedCLP >= 0
21              @userHasEnoughMoneyToDoTheTransaction = true
22            else
23              @userHasEnoughMoneyToDoTheTransaction = false
24            end
25
26
27            if @userHasEnoughMoneyToDoTheTransaction
28              if @transaction.save
```

Ejemplo cambios

BuyController > Método “create”:

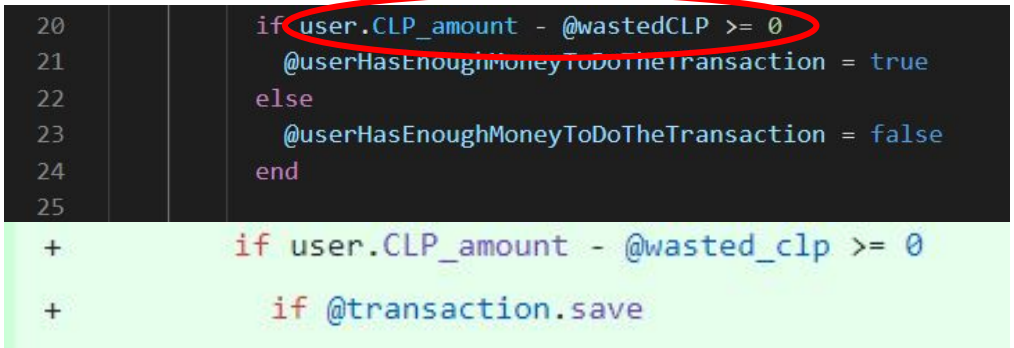
- Single-responsibility principle
- Bloaters - Long Method
- Variables innecesarias



Ejemplo cambios

BuyController > Método “create”:

- Single-responsibility principle
- Bloaters - Long Method
- Variables innecesarias



Ejemplo cambios

Buy controller > Método “create”:

- Separación en varios métodos
- métodos más cortos

buy_controller.rb

```
1   class BuyController < ApplicationController
2       protect_from_forgery unless: -> { request.format.json? || request.format.xml? }
3   >   def create ···
13      end
14
15  >   def aux_create ···
31      end
32
33  >   def aux_generate_transaction(user, amount) ···
43      end
44
45  >   def aux_transaction_save(user, unique_bit_fake_currency, wasted_clp, amount) ···
53      end
54  end
```

#Comentarios

```
+   # Take all transactions of the user in the API and see which transactions
+   # we havent review since the date of the last charge (queried in db), then we update
    def make_it_charge
```

```
+   # Aux function to generate a transactions when we want to buy
    def aux_generate_buy_transaction
```

Funciones explicadas

```
-     # Este código se debería correr solamente al inicio,
-     # cuando setea los valores de compra y venta del BTF
+     # This code should be run just at the begining,
+     # when it setups the values of buy and sell of BTF
      if Bitfake.all.count < 1
```

Ser consecuentes en el idioma

Flujo más relevante

Implementación Cronjob

Calidad de código

Refactoring

Conclusiones

Conclusiones

1. Herramientas para verificar la calidad del código 
2. Mantener las buenas prácticas ayuda a entender el código cuando externos revisan. 
3. Mantener un código alineado con el objetivo del proyecto (en términos de la semántica) es muy relevante

Bank twelve - Exchange Four

Grupo 24

Felipe Cabrera - Martín Cadagan - Camila Chávez - Mariana Ortega - Joaquín Ossandón