

PRESENTACIÓN FINAL

Grupo 17

Daniel Estay
Javiera Ferrada
Simón Garcés
Francisco Paniagua
Antonia Soto

PULL REQUEST

- Uso de plantilla
- Instancia para revisar y mejorar detalles de implementación
- Generar discusión para un código de buena calidad

feat: bank account model and functions #4

 Merged absoto4 merged 7 commits into [feature/users](#) from [accounts](#) on 21 Oct

 Conversation 11  Commits 7  Checks 0  Files changed 31



JaviFerrada commented on 21 Oct

Description

Bank Account Model and Controllers. Includes CRUD and definition of functions(investment,withdrawal,invest, transfer, getAlltransactions, getInvestmentBalance). Nested routes users - bank_account.

Type of change

Please delete options that are not relevant.

- ☒ New feature (non-breaking change which adds functionality)

How Has This Been Tested?

Please describe the tests that you ran to verify your changes (Like web browser's or mobile devices). Provide instructions so we can reproduce.

- ☒ [X] Chrome Desktop
- ☐ Safari Desktop
- ☐ Chrome Mobile (not desktop emulator)
- ☐ iOS Mobile (not desktop emulator)

Before this PR

After this PR

Bank Account: model, controllers, routes
Views for index: to show resume balance and transactions
Relation User has_one BankAccount
View to Activate(create) new account for user without BankAccounts
BankAccount belongs_to User

Checklist:

- ☒ [X] My code follows the style guidelines of this project
- ☒ [X] I have performed a self-review of my own code
- ☒ My changes generate no new warnings
- ☒ [X] The target of this commit is 'dev'



sfga reviewed on 21 Oct

[View changes](#)

app/controllers/application_controller.rb **Outdated**

```
... .. @@ -6,6 +6,10 @@ class ApplicationController < ActionController::Base
6 6 private
7 7
8 8 def after_sign_in_path_for(resource)
9 - user_path(resource)
9 + if BankAccount.where(user_id: resource.id).length == 1
```



sfga on 21 Oct

😊 ...

la consulta se puede simplificar a:
`if resource.bank_account`



sfga on 21 Oct

😊 ...

PD: de todas maneras, buen acierto en forzar la redirección para que el nuevo usuario cree la cuenta del banco



Reply...

[Resolve conversation](#)



sfga reviewed on 21 Oct

[View changes](#)

app/controllers/bank_accounts_controller.rb **Outdated**

```
2 +
3 +
4 +
5 + def index
```



sfga on 21 Oct

😊 ...

Un usuario tiene a lo más una cuenta, por lo que esta acción correspondería al `show` (el `index` debería estar, al menos, restringido para usuarios normales)



Reply...

[Resolve conversation](#)



sfga reviewed on 21 Oct

[View changes](#)

app/controllers/bank_accounts_controller.rb **Outdated**

```
16 + @bank_account = BankAccount.create(account_balance: rand(10000..100000), investment_balance: 0.0, us
17 + @bank_account.account_number = assign_unique_account_number(@bank_account)
18 + @bank_account.save
19 + redirect_to user_bank_accounts_path(@user)
```



sfga on 21 Oct

😊 ...

debería dirigir a `user_bank_account_path` (acción `show`)



Reply...

[Resolve conversation](#)



sfga reviewed on 21 Oct

[View changes](#)

app/controllers/pages_controller.rb **Outdated**

```
5 - def home; end
5 + def home
6 + @user = current_user
7 + if @user
```



sfga on 21 Oct

😊 ...

bastaría solo con `if current_user`



Reply...

[Resolve conversation](#)

sfga reviewed on 21 Oct

View changes

```
app/views/bank_accounts/index.html.erb Outdated
1 + <% if current_user %>
2 +
3 + <h1>Bank Account</h1>
4 + <% bank_account = @bank_accounts[0] %>
```

sfga on 21 Oct

😊 ...

al ocupar `show`, no sería necesaria obtener el primer y unico elemento de la colección de `bank_accounts` del usuario
además, lo ideal es que la lógica de asignaciones quede en el controlador y no en la vista



Reply...

Resolve conversation

sfga reviewed on 21 Oct

View changes

```
app/views/users/show.html.erb
20 - </p>
1 + <h1> Your account </h1>
2 +
3 + <%= Agregar resumen de cuentas %>
```

sfga on 21 Oct

😊 ...

esta página esta disponible en caso que el usuario no active su cuenta, por lo que sería útil que existiese un bloque if para diferenciar el contenido mostrado para un usuario con cuenta y otro usuario sin cuenta



Reply...

Resolve conversation

sfga reviewed on 21 Oct

View changes

```
app/controllers/bank_accounts_controller.rb
... .. @@ -0,0 +1,72 @@
1 + class BankAccountsController < InheritedResources::Base
```

sfga on 21 Oct

😊 ...

Para el futuro deberían setearse algunos `before_action` (revisa el controlador de users, podría ser algo parecido a eso) para evitar que otros usuari@s (logueados o no, con excepción de usuari@s admin) puedan realizar alguna accion sobre el contenido de una cuenta que no les pertenece.



Reply...

Resolve conversation

sfga reviewed on 21 Oct


View changes











sfga left a comment

😊 ...

Buen trabajo en general ✓

- La acción `index` debería convertirse en la acción `show` para `bank_account`
- Dejó algunas sugerencias de corrección (algunas de ellas las voy a dejar hechas, con el objetivo de tener disponible estos cambios en dev para l@s avance que hagan los demás)

 **sfga** added 5 commits on 21 Oct

-   fix: apply changes suggested by rubocop 3b280de
-   fix: tiny refactor on referencing variables b49d0c6
-   fix: add user reference on bank_accounts table c0e8215
-   fix: transfer bank accounts controller index action code into show ac... 0182c45
-   fix: add conditional scope to differentiate behavior for users with ac... 9f38fa0



sfga commented on 21 Oct • edited ▾



@JaviFerrada

Subí algunos cambios (correcciones basadas en los comentarios del review que hice más arriba + correcciones de `rubocop`), por lo que sugiero:

- revisar si existen dudas sobre los nuevos cambios
- cambiar el destino del merge hacia la rama `dev`, en caso que el PR #3 (`feature/users` -> `dev`) se apruebe y se realice antes que este PR, dado que el primero de los commits "nuevos" corresponde a los cambios bajo revisión en #3.



JaviFerrada commented on 21 Oct

Author 

@sfga Revise los cambios y me parecen muy buenos, gracias! Lo probe y funciona correctamente

BUENAS PRÁCTICAS

```
def set_bank_account
  @bank_account = BankAccount.find(params[:id])
end
```

app/controllers/bank_accounts_controller.rb

```
...    @@ -96,4 +99,21 @@ def assign_unique_account_number(bank_account)
96      99      def six_digit_rand
97      100      (SecureRandom.random_number(9e5) + 1e5).to_i
98      101      end
102    +
103    +      def require_permission
104    +        @bank_account = BankAccount.find(params[:id])
```



sfga on Oct 30



recomiendo utilizar una funcion auxiliar (set_bank_account por ejemplo) para usar como before_action: que setee la cuenta (i.e bank_account = BankAccount.find(params[:id]) ... pq se esta repitiendo demasiadas veces esa línea

Uso de *Before_action* para evitar pedir la cuenta del usuario actual en cada método.

BUENAS PRÁCTICAS

Validaciones desde el modelo y no desde el controlador (evitar bloating).

```
class User < ApplicationRecord
  has_one :bank_account, dependent: :destroy
  devise :database_authenticatable, :registerable,
         :recoverable, :rememberable, :validatable

  validates :email, presence: true, uniqueness: true
  validates :encrypted_password, presence: true
  validates :firstname, presence: true
  validates :lastname, presence: true
  validates :age, numericality: { greater_than_or_equal_to: 18 }
  validates :rut, presence: true, uniqueness: true
  validates :rut, format: { with: /\A^\d{6,8}[-][0-9kK]{1}$\z/,
                           message:
                             ""RUT solo debe contener entre 6 y 8 dígitos continuados
                             de un guión y el dígito verificador. No agregue espacios ni
                             puntos separadores."" }

  validate :validate_rut
  before_save :generate_token
```

BUENAS PRÁCTICAS

Validaciones desde el modelo y no desde el controlador (evitar bloating).

```
def validate_rut
  correct_dv = calculate_dv(rut[0..-3])
  compared_dv = rut[-1]
  dv_correcto = correct_dv.eql? compared_dv
  unless dv_correcto
    errors[:base] << "RUT es incorrecto"
  end
end

def calculate_dv(base_rut)
  reversed_digits = base_rut.reverse.chars.map(&:to_i)
  factors = (2..7).to_a.cycle
  suma = reversed_digits.zip(factors).inject(0) { |sum, (d, f)| sum + d * f }
  (-suma) % 11 == 10 ? 'K' : ((-suma) % 11).to_s
end

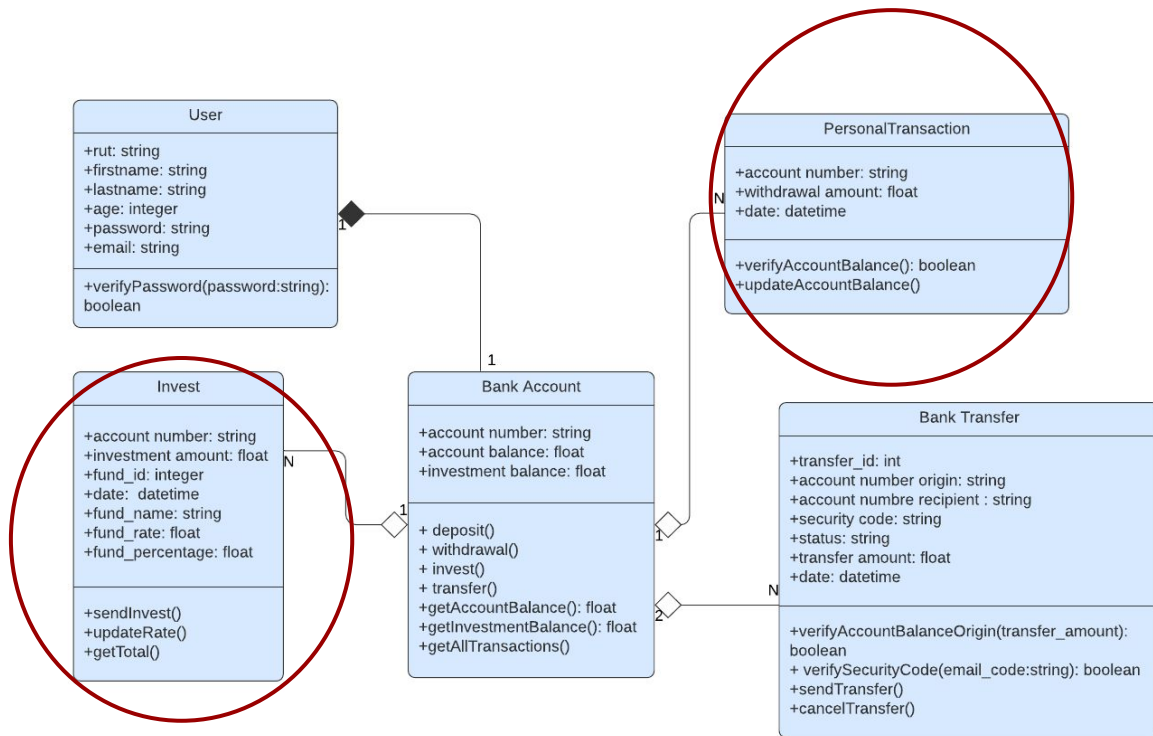
def generate_token
  self.token ||= SecureRandom.urlsafe_base64
end
```

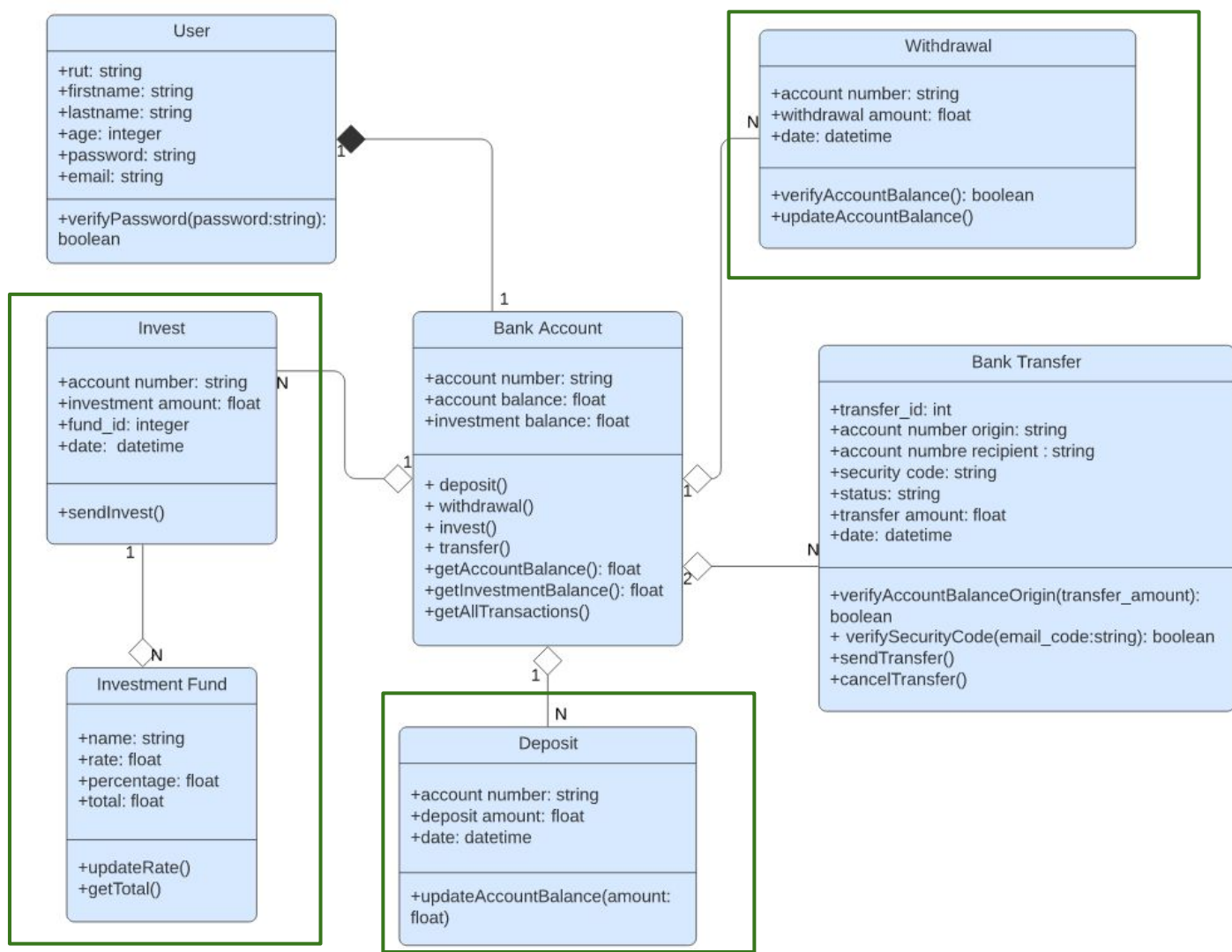

BUENAS PRÁCTICAS

Importancia de la etapa de diseño

Separación de responsabilidades:

- Invest e InvestmentFund
- Deposit y Withdrawal



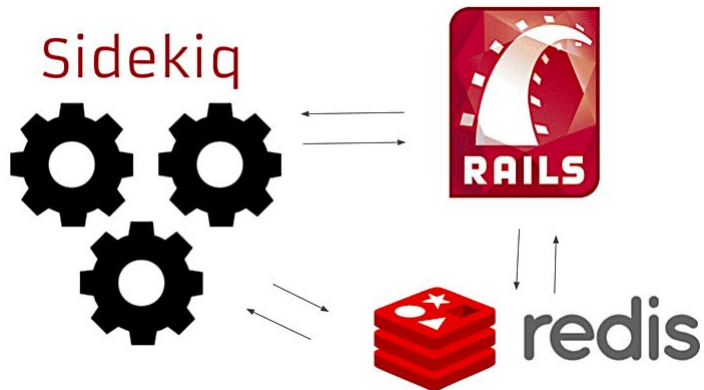


BUENAS PRÁCTICAS

Separación de responsabilidades bajo los principios SOLID



Tareas secundarias realizadas **separadamente** del flujo principal (y de forma eficiente si es posible)



BUENAS PRÁCTICAS

Entrega 2 (Banco):

Mailer de forma
síncrona

Afecta el flujo
principal de la
aplicación



BUENAS PRÁCTICAS

Entrega 2 (Banco):

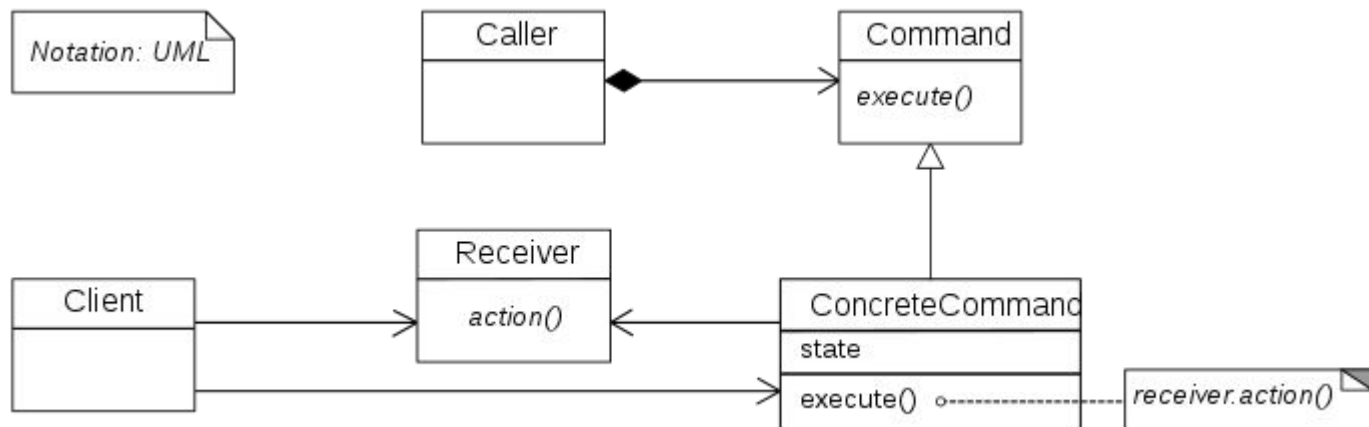
Mailer ejecutado de forma asíncrona

Mejor UX!



PATRÓN DE DISEÑO

E3 - Exchange: Uso del patrón Command para actualizar el valor de la BitFake.



PATRÓN DE DISEÑO

E3 - Exchange: Uso del patrón Command para actualizar el valor de la BitFake.

app > commands >  update_bitfake_price.rb

```
You, 4 minutes ago | 1 author (You)
1 module UpdateBitfakePrice
2   def update_bitfake_price
3     bitfakes = Bitfake.all
4     bitfakes.each(&:update_price)
5   end
6 end
7
8
9
```

lib > tasks >  cron_tasks.rake

```
You, a few seconds ago | 2 authors (You and others)
1 namespace :crontasks do
2   desc 'Scheduled tasks'
3
4   task update_bitfake_price: :environment do
5     include UpdateBitfakePrice
6     loop do
7       update_bitfake_price
8       break unless Rails.env.development? || Rails.env.test?
9
10      sleep 3600 * 3
11    end
12  end
13 end
```

¡Gracias por la atención!

Referencias

Contenido:

- Balart, F. *Services, Commands y otros poderosos patrones en Rails*. Código Banana: El blog de Platanus. Disponible en: <https://blog.platan.us/services-commands-y-otros-poderosos-patrones-en-rails-27c2d3aa7c2e>. Consultado el 2 de diciembre de 2020
- Refactoring.Guru. *Command Design Pattern*. Disponible en: <https://refactoring.guru/es/design-patterns/command>. Consultado el 2 de diciembre de 2020

Imágenes

- Diapositiva 11
 - Brito, R. *Upgrading from DelayedJob to Sidekiq in a mature Rails webapp*. Goiabada Blog. Disponible en: <https://goiabada.blog/upgrading-from-delayedjob-to-sidekiq-in-a-mature-rails-webapp-966fb0d11ace>. Consultado el 3 de diciembre de 2020.
- Diapositivas 12-13:
 - Gebhardt, D. *Tame Rails Email Dragons with Mailhopper*. Cerebris Blog. Disponible en: <https://www.cerebris.com/blog/2011/09/07/tame-rails-email-dragons-with-mailhopper/>. Consultado el 2 de diciembre de 2020
- Diapositiva 14:
 - Wikimedia. *UML diagram of the command pattern*. Disponible en: https://upload.wikimedia.org/wikipedia/commons/b/bf/Command_pattern.svg Consultado el 3 de diciembre de 2020.