

# Presentación Final

## IIC2113 - Diseño Detallado de Software



Integrantes  
Javiera Inostroza  
Juan Lopez  
Manuel Muñoz  
Samuel Zúñiga  
Verner Codoceo


Fecha: 4/12/2020

Profesora: María Fernanda Sepúlveda

# Destacado: Arbitraje

```
1 class GFundeventJob < ApplicationJob
2   queue_as :default
3
4   def perform(*_args)
5     puts 'Checkeando arbitraje...'
6     api = ExchangeApi.instance
7     api.login
8
9     prices = api.prices("BTF")
10    if prices["selling_price"] < prices["buying_price"]
11      puts 'Arbitrando...'
12      gain = arbitrate(api, prices)
13
14      risk_fund = Fund.find_by(name:
15        "Fondo de Inversión: Riesgo Alto")
16      increase_fund(risk_fund, gain)
17    else
18      puts "No es posible arbitrar"
19    end
20
21    api.logout
22  end
23
24  def arbitrate(api, prices)
25  end
26
27  def increase_fund(fund, increase_amount)
28  end
29 end
```


# Singleton: ExchangeApi



```
1  require 'singleton'
2  require 'json'
3
4  class ExchangeApi
5    include Singleton
6
7    def initialize
8      @email = ENV["EXCHANGE_API_EMAIL"]
9      @password = ENV["EXCHANGE_API_PASSWORD"]
10     @token = nil
11     @http_client = HttpClient.instance
12   end
13
14   ...
15
16 end
```

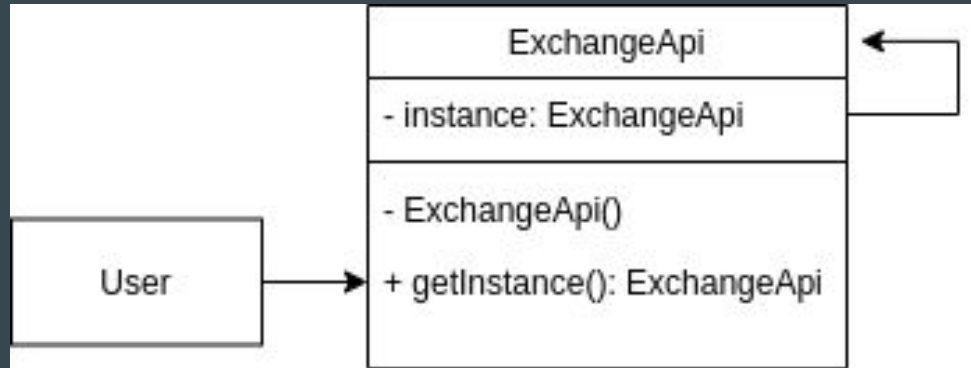
**DEMO**

# Singleton: ExchangeApi



```
1  require 'singleton'
2  require 'json'
3
4  class ExchangeApi
5    include Singleton
6
7    def initialize
8      @email = ENV["EXCHANGE_API_EMAIL"]
9      @password = ENV["EXCHANGE_API_PASSWORD"]
10     @token = nil
11     @http_client = HttpClient.instance
12   end
13
14   ...
15
16 end
```

# Singleton: ExchangeApi



# Refactor



```
1 def prices(cryptocurrency_name)
2   begin
3     price_url = @base_url + '/cryptocurrencies'
4     response = RestClient.get(price_url, headers)
5     result = JSON.parse(response.body)
6
7     cryptocurrency_prices = result["cryptocurrencies"].detect do |crypto|
8       crypto["name"] == cryptocurrency_name
9     end
10    cryptocurrency_prices
11  rescue RestClient::Exceptions::OpenTimeout
12    prices(cryptocurrency_name)
13  end
14 end
```



```
1 def prices(cryptocurrency_name)
2   response = get_request('/cryptocurrencies', headers)
3   result = JSON.parse(response.body)
4
5   cryptocurrency_prices = result["cryptocurrencies"].detect do |crypto|
6     crypto["name"] == cryptocurrency_name
7   end
8   cryptocurrency_prices
9 end
```

# Singleton: HttpClient



```
1  require 'rest-client'
2  require 'singleton'
3
4  class HttpClient
5    include Singleton
6
7    def initialize
8      @base_url = ENV["EXCHANGE_API_URL"]
9    end
10
11    def get_request(url, headers)
12    end
13
14    def post_request(url, body, headers=nil)
15    end
16
17  end
```



# Presentación Final

## IIC2113 - Diseño Detallado de Software



Integrantes  
Javiera Inostroza  
Juan Lopez  
Manuel Muñoz  
Samuel Zúñiga  
Verner Codoceo

Fecha: 4/12/2020

Profesora: María Fernanda Sepúlveda