

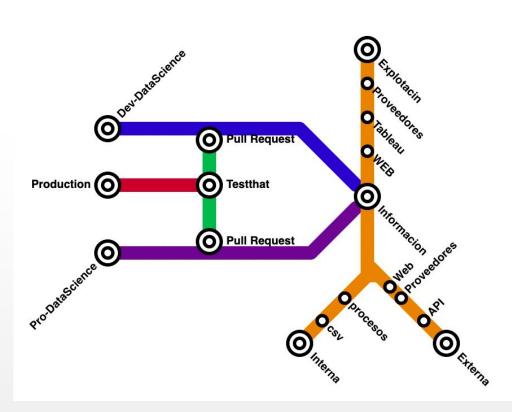


R en la empresa: cómo vender más disfraces integrando R.

- 1 Nuestros entornos
- 2 L1: Desarrollo, validación y puesta en producción
- 3 L2: Algoritmo F-Score para ordenación de productos
- 4 L3: Gestión inteligente de campañas en Google Ads



1 > Nuestros entornos



Dev > 32 Gb Ram > 100 Gb ROM Cloud + VM

8-18 Núcleos

Git

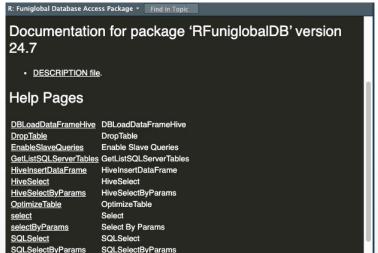
Tableau /
Shiny
MySQL,

MongoDB, PostgreSQL



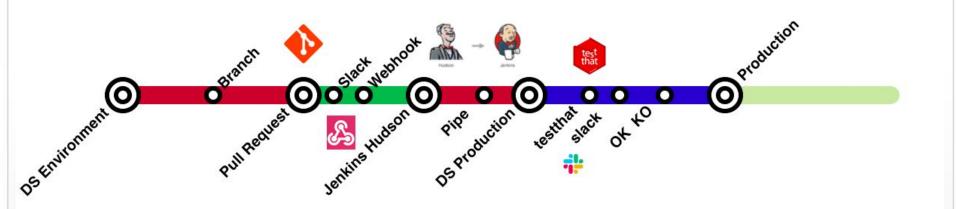


cat("INSERCIÓN EN LA BBDD\n")
SetupDBConnection("conexion1")
TruncateTable("aggregated_sales_shop_pending", "datascience")
DBLoadDataFrame(pending.sales.df, "aggregated_sales_shop_pending", "datascience")
OptimizeTable("aggregated_sales_shop_pending", "datascience")





L1: Desarrollo, validación y puesta en producción



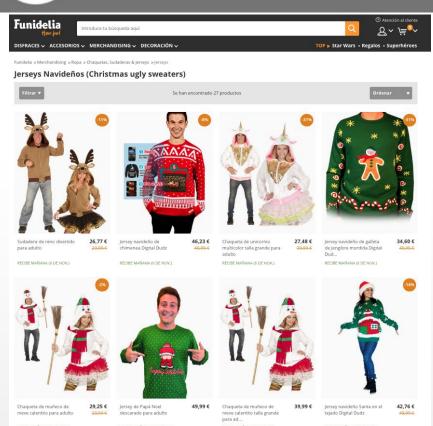


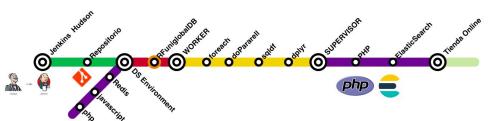
2.2 Ll: Desarrollo, validación y puesta en producción

```
#test unitario
test that ("Calculate Royalties OUT: vector expected", {
  expect_equal(length(CalculateAdjustedRoyaltiesDevos(LoadRoyaltiesFakeData(1))), 3)
  expect equal(length(CalculateAdjustedRoyaltiesDevos(LoadRoyaltiesFakeData(2))), 2)
  expect equal(length(CalculateAdjustedRoyaltiesDevos(LoadRoyaltiesFakeData(3))), 1)
  expect equal(length(CalculateAdjustedRoyaltiesDevos(LoadRoyaltiesFakeData(4))), 4)
 expect equal(length(CalculateAdjustedRoyaltiesDevos(LoadRoyaltiesFakeData(5))), 10)
})
cat("Calculate Royalties OUT: vector expected: .... OK\n")
         Coste - Pedido: Royalties
                     I coste-pedido-testthatCalculate Royalties OUT: object type: .... OK
         Calculate Royalties OUT: vector expected: ..... OK
         Calculate Royalties OUT: value expected: ..... OK
         / I 16
                     I coste-pedido-testthat
         — Results —
         OK:
         Failed:
         Warnings: 0
         Skipped: 0
```

3

L2: Algoritmo F-Score para ordenación de productos







3.2 L2: Algoritmo F-Score para ordenación de productos



VS



Tiempo ddply: 204.936 sec elapsed

sgldf: 47.707 sec elapsed

```
tic("Tiempo sqldf")
sqldf.df <- sqldf("select id_shop, id_landing, count(1) as productos,</pre>
                          count(distinct(ds_etiqueta_seller)) as etiquetas,
                          avg(if(id_shop = 1, fscore, null)) as esp
                          from 'datos.df'
                          group by id_shop, id_landing", drv = "SOLite")
toc()
tic("Tiempo ddply")
ddply.df <- ddply(datos.df, c("id_shop", "id_landing"), summarise,
                  productos = length(id_product),
                   etiquetas = length(unique(ds_etiqueta_seller)),
                   esp = mean(fscore[id_shop == 1]))
toc()
```



L3: Gestión inteligente de campañas en Google Ads

