Modelo_Proyecto

Arturo Camacho

2026-01-30

Primer intento del proyecto

```
# ============================================================
#   MODELO DE RIESGO DE SALUD MENTAL COMBINANDO DOS DATASETS
# ============================================================

library(tidyverse)

## ── Attaching core tidyverse packages ────────────────────────
tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.1     ✓ stringr   1.5.2
## ✓ ggplot2   4.0.0     ✓ tibble    3.3.0
## ✓ lubridate 1.9.4     ✓ tidyr     1.3.1
## ✓ purrr     1.1.0
## ── Conflicts ──────────────────────────────────────────
tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force
all conflicts to become errors

library(janitor)

##
## Adjuntando el paquete: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test

library(caret)

## Warning: package 'caret' was built under R version 4.5.2

## Cargando paquete requerido: lattice
##
## Adjuntando el paquete: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift

library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.5.2

## randomForest 4.7-1.2
## Type rfNews() to see new features/changes/bug fixes.
##
## Adjuntando el paquete: 'randomForest'
##
## The following object is masked from 'package:dplyr':
##
##     combine
##
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
# ============================================================
# 1. CARGA CORRECTA DE LOS DATASETS
# ============================================================

# Survey (separado por comas)
survey <- read_csv("survey.csv") %>% clean_names()
```

```
## Rows: 1259 Columns: 27
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (25): Gender, Country, state, self_employed, family_history,
treatment,...
## dbl   (1): Age
## dttm  (1): Timestamp
##
## ℹ Use `spec()` to retrieve the full column specification for this
data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet
this message.
```

```
# Absenteeism (separado por punto y coma)
absent <- read_delim(
  "Absenteeism_at_work.csv",
  delim = ";",
  col_names = TRUE
) %>% clean_names()
```

```
## Rows: 740 Columns: 21
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ";"
## dbl (21): ID, Reason for absence, Month of absence, Day of the week,
Seasons...
##
## ℹ Use `spec()` to retrieve the full column specification for this
```

```
data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet
this message.

# ============================================================
# 2. CREAR VARIABLE OBJETIVO (RIESGO)
# ============================================================

survey <- survey %>%
  mutate(
    risk = case_when(
      work_interfere %in% c("Often", "Sometimes") ~ 1,
      TRUE ~ 0
    )
  )

# ============================================================
# 3. CREAR ID ARTIFICIAL PARA COMBINAR
# ============================================================

min_rows <- min(nrow(survey), nrow(absent))

survey <- survey %>% slice(1:min_rows) %>% mutate(id = 1:min_rows)
absent <- absent %>% slice(1:min_rows) %>% mutate(id = 1:min_rows)

# ============================================================
# 4. COMBINAR DATASETS
# ============================================================

combined <- left_join(survey, absent, by = "id")

# ============================================================
# 5. SELECCIÓN DE VARIABLES PREDICTORAS
# ============================================================

predictoras <- combined %>%
  select(
    risk,
    # Variables psicológicas
    family_history, benefits, anonymity, supervisor, coworkers,
    mental_health_consequence, phys_health_consequence,
    mental_vs_physical, leave,

    # Variables de ausentismo
    absenteeism_time_in_hours, distance_from_residence_to_work,
    service_time, social_drinker, social_smoker, disciplinary_failure,
    body_mass_index, reason_for_absence, month_of_absence,

    # Demográficas
    age.y
  )

# ============================================================
# 6. LIMPIEZA Y TRANSFORMACIÓN
# ============================================================

# Convertir categóricas a factor
predictoras <- predictoras %>%
  mutate(across(where(is.character), as.factor))

# Imputación simple
for (col in names(predictoras)) {
  if (is.factor(predictoras[[col]])) {
    moda <- names(sort(table(predictoras[[col]]), decreasing = TRUE))[1]
    predictoras[[col]][is.na(predictoras[[col]])] <- moda
  }
}

predictoras$age[is.na(predictoras$age)] <- median(predictoras$age, na.rm
= TRUE)
```

```
## Warning: Unknown or uninitialised column: `age`.
```

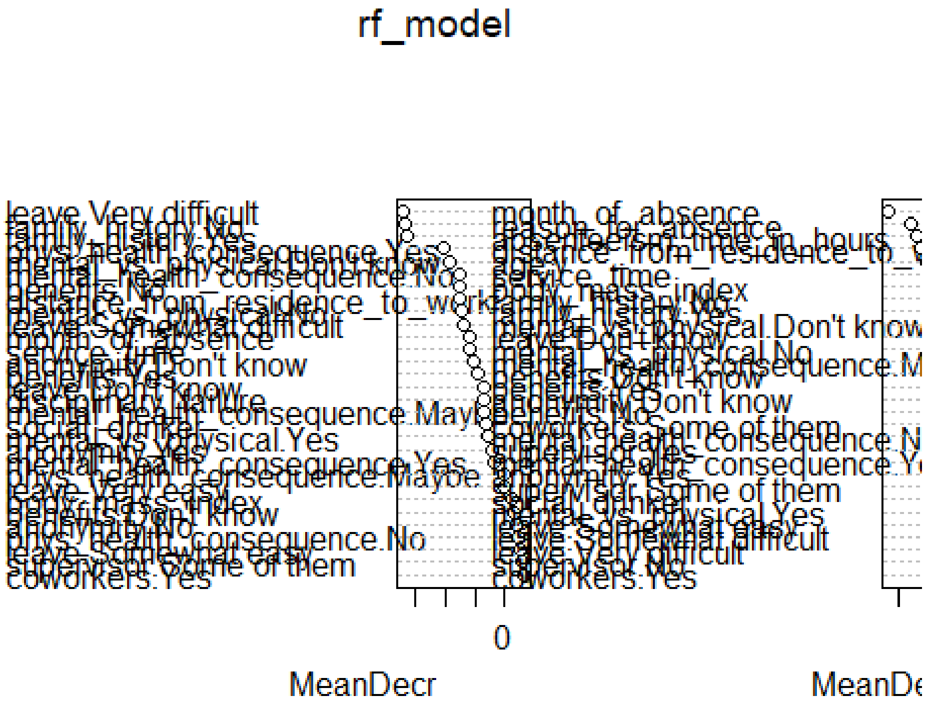
```
## Warning: Unknown or uninitialised column: `age`.
## Unknown or uninitialised column: `age`.
```

```
# ============================================================
# 7. CODIFICACIÓN ONE-HOT
# ============================================================

dummies <- dummyVars(risk ~ ., data = predictoras)
X <- predict(dummies, newdata = predictoras) %>% as.data.frame()
y <- predictoras$risk

# ============================================================
# 8. DIVISIÓN TRAIN / TEST
# ============================================================

set.seed(123)
trainIndex <- createDataPartition(y, p = 0.7, list = FALSE)
X_train <- X[trainIndex, ]
X_test  <- X[-trainIndex, ]
y_train <- y[trainIndex]
y_test  <- y[-trainIndex]

# ============================================================
# 9. ENTRENAR MODELO RANDOM FOREST
# ============================================================
```

```r
rf_model <- randomForest(
  x = X_train,
  y = as.factor(y_train),
  ntree = 300,
  importance = TRUE
)

# ============================================================
# 10. EVALUACIÓN DEL MODELO
# ============================================================

pred <- predict(rf_model, X_test)
confusionMatrix(pred, as.factor(y_test))
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction  0  1
##          0 84 41
##          1 31 66
##                                           
##                Accuracy : 0.6757          
##                  95% CI : (0.6098, 0.7368)
##     No Information Rate : 0.518           
##     P-Value [Acc > NIR] : 1.376e-06       
##                                           
##                   Kappa : 0.3484          
##                                           
##  Mcnemar's Test P-Value : 0.2888          
##                                           
##             Sensitivity : 0.7304          
##             Specificity : 0.6168          
##          Pos Pred Value : 0.6720          
##          Neg Pred Value : 0.6804          
##              Prevalence : 0.5180          
##          Detection Rate : 0.3784          
##    Detection Prevalence : 0.5631          
##       Balanced Accuracy : 0.6736          
##                                           
##        'Positive' Class : 0               
## 
```

```r
# ============================================================
# 11. IMPORTANCIA DE VARIABLES (KPIs)
# ============================================================

varImpPlot(rf_model)
```



```
importance(rf_model)

##                                                 0           1
MeanDecreaseAccuracy
## family_history.No                     6.019296151  4.06286608
6.66924477
## family_history.Yes                    5.949033651  3.33671921
6.48097062
## benefits.Don't know                  -0.289928228 -0.82975839           -
0.66710891
## benefits.No                           4.079417041 -0.01761710
3.06335781
## benefits.Yes                          2.553628717 -0.47842428
1.86324978
## anonymity.Don't know                  1.478023180  1.33054802
2.12071313
## anonymity.No                         -2.018588136  0.69824441           -
0.73706741
## anonymity.Yes                        -0.122085892  1.48403974
0.91132055
## supervisor.No                        -0.503243958 -3.26339173           -
2.63370767
## supervisor.Some of them              -0.006833788 -1.66835871           -
1.18558238
## supervisor.Yes                        1.750875241 -5.19382744           -
1.80603646
## coworkers.No                          1.126358571 -3.39932800           -
```

```
1.53378224
## coworkers.Some of them          -0.996369141 -2.43487439          -
2.18959367
## coworkers.Yes                   -1.967442743  0.41743865          -
1.32379912
## mental_health_consequence.Maybe  1.531191714  0.27315446
1.41040288
## mental_health_consequence.No     5.180213926 -2.77807340
3.07721214
## mental_health_consequence.Yes    0.362440655  0.72070958
0.76407661
## phys_health_consequence.Maybe    0.252881920 -0.09748453
0.12554993
## phys_health_consequence.No      -0.201793645 -1.25097008          -
0.87796748
## phys_health_consequence.Yes      3.160504296  3.32962341
4.11154410
## mental_vs_physical.Don't know    2.224138213  3.09695743
3.66789460
## mental_vs_physical.No            2.419965209  1.58532854
2.95004871
## mental_vs_physical.Yes           2.985047876 -1.73999185
1.09368286
## leave.Don't know                 0.145765302  2.16108525
1.46978314
## leave.Somewhat difficult         4.449315487 -0.45657311
2.72951030
## leave.Somewhat easy             -0.267906619 -1.35790968          -
1.01062370
## leave.Very difficult             3.830668174  6.07382059
6.85574741
## leave.Very easy                  1.835699359 -1.79316469
0.05662196
## absenteeism_time_in_hours       -1.012925478 -1.33586491          -
1.62579282
## distance_from_residence_to_work  3.357988592  0.52399458
3.03494647
## service_time                     3.443732814 -0.63839025
2.39489676
## social_drinker                   3.121935172 -1.77204080
1.30294245
## social_smoker                   -2.963150162  0.53027053          -
2.28289238
## disciplinary_failure             1.228859942  0.84048074
1.43884320
## body_mass_index                 -0.649066638  0.07467657          -
0.49081614
## reason_for_absence              -4.583115174 -0.42226667          -
3.69507964
## month_of_absence                 3.476095276 -0.20579126
```

```
2.41775379
## age.y                           -0.798974912 -4.42809036          -
3.39164864
##                                 MeanDecreaseGini
## family_history.No                       8.696583
## family_history.Yes                      8.160438
## benefits.Don't know                     4.998129
## benefits.No                             4.753393
## benefits.Yes                            4.994814
## anonymity.Don't know                    4.973531
## anonymity.No                            1.482109
## anonymity.Yes                           4.197358
## supervisor.No                           3.624904
## supervisor.Some of them                 4.184087
## supervisor.Yes                          4.432824
## coworkers.No                            3.185757
## coworkers.Some of them                  4.698673
## coworkers.Yes                           3.567315
## mental_health_consequence.Maybe         5.016269
## mental_health_consequence.No            4.644955
## mental_health_consequence.Yes           4.222681
## phys_health_consequence.Maybe           3.172421
## phys_health_consequence.No              3.353583
## phys_health_consequence.Yes             2.455543
## mental_vs_physical.Don't know           5.979049
## mental_vs_physical.No                   5.079622
## mental_vs_physical.Yes                  3.840446
## leave.Don't know                        5.666819
## leave.Somewhat difficult                3.770126
## leave.Somewhat easy                     3.801121
## leave.Very difficult                    3.670412
## leave.Very easy                         3.528954
## absenteeism_time_in_hours              16.633883
## distance_from_residence_to_work        16.035580
## service_time                           14.325050
## social_drinker                          4.180914
## social_smoker                           1.935921
## disciplinary_failure                    1.651011
## body_mass_index                        13.483526
## reason_for_absence                     17.645785
## month_of_absence                       21.855071
## age.y                                  14.760704
```

Segundo Intento del Modelo

```
library(tidyverse)
library(janitor)
library(caret)
library(randomForest)
```

```r
# Cargar survey
survey <- read_csv("survey.csv") %>% clean_names()

## Rows: 1259 Columns: 27
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (25): Gender, Country, state, self_employed, family_history,
treatment,...
## dbl   (1): Age
## dttm  (1): Timestamp
##
## ℹ Use `spec()` to retrieve the full column specification for this
data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet
this message.

# Cargar absenteeism (separado por ;)
absent <- read_delim(
  "Absenteeism_at_work.csv",
  delim = ";",
  col_names = TRUE
) %>% clean_names()

## Rows: 740 Columns: 21
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ";"
## dbl (21): ID, Reason for absence, Month of absence, Day of the week,
Seasons...
##
## ℹ Use `spec()` to retrieve the full column specification for this
data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet
this message.

# Limpieza profesional de gender
survey <- survey %>%
  mutate(
    gender = str_to_lower(gender),
    gender = case_when(
      str_detect(gender, "male|man|cis male|cis-man|m\\b") ~ "male",
      str_detect(gender, "female|woman|cis female|cis-woman|f\\b") ~
"female",
      TRUE ~ "nonbinary"
    ),
    gender = as.factor(gender)
  )

# Crear variable objetivo (riesgo)
```

```r
survey <- survey %>%
  mutate(
    risk = case_when(
      work_interfere %in% c("Often", "Sometimes") ~ 1,
      TRUE ~ 0
    )
  )

# Crear ID artificial correctamente
survey <- survey %>%
  mutate(id = row_number()) %>%   # aquí sí se puede usar row_number()
  select(
    id,
    risk,
    gender,
    family_history, benefits, anonymity, supervisor, coworkers,
    mental_health_consequence, phys_health_consequence,
    mental_vs_physical, leave,
    age
  )

absent <- absent %>%
  select(
    id,
    reason_for_absence,
    month_of_absence,
    distance_from_residence_to_work,
    service_time,
    social_drinker,
    social_smoker,
    disciplinary_failure,
    body_mass_index,
    absenteeism_time_in_hours,
    pet,
    son,
    age
  )

min_rows <- min(nrow(survey), nrow(absent))

survey <- survey %>% slice(1:min_rows)
absent <- absent %>% slice(1:min_rows)

combined <- left_join(survey, absent, by = "id")

predictoras <- combined %>%
  select(
    risk,
    gender,
    family_history, benefits, anonymity, supervisor, coworkers,
    mental_health_consequence, phys_health_consequence,
    mental_vs_physical, leave,
    reason_for_absence, month_of_absence,
    distance_from_residence_to_work, service_time,
    social_drinker, social_smoker, disciplinary_failure,
    body_mass_index, absenteeism_time_in_hours,
    pet, son,
    age.y
  ) %>%
  rename(age = age.y)

# ============================================================
# 7. IMPUTACIÓN DEFINITIVA Y NORMALIZACIÓN
# ============================================================

# 1. Convertir categóricas a factor
predictoras <- predictoras %>%
  mutate(across(where(is.character), as.factor))

# 2. Imputación para factores (moda)
for (col in names(predictoras)) {
  if (is.factor(predictoras[[col]])) {
    moda <- names(sort(table(predictoras[[col]]), decreasing = TRUE))[1]
    predictoras[[col]][is.na(predictoras[[col]])] <- moda
  }
}

# 3. Imputación para numéricas (mediana)
predictoras <- predictoras %>%
  mutate(across(
    where(is.numeric),
    ~ ifelse(is.na(.), median(., na.rm = TRUE), .)
  ))

# 4. Normalización SOLO después de imputar
predictoras <- predictoras %>%
  mutate(across(
    c(distance_from_residence_to_work, service_time,
      body_mass_index, absenteeism_time_in_hours,
      pet, son, age),
    ~ scale(.) %>% as.numeric()
  ))

dummies <- dummyVars(risk ~ ., data = predictoras)
X <- predict(dummies, newdata = predictoras) %>% as.data.frame()
y <- predictoras$risk

set.seed(123)
trainIndex <- createDataPartition(y, p = 0.7, list = FALSE)
X_train <- X[trainIndex, ]
X_test  <- X[-trainIndex, ]
```

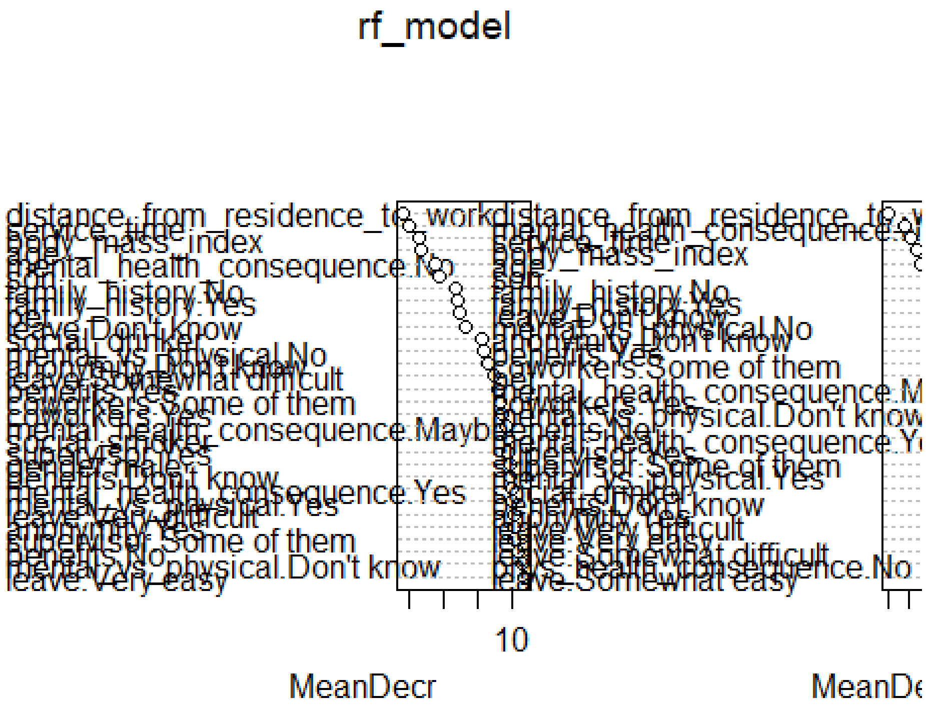
```
y_train <- y[trainIndex]
y_test  <- y[-trainIndex]

# ============================================================
# DIAGNÓSTICO DE NA
# ============================================================

colSums(is.na(X_train))
```

```
##                   gender.female                     gender.male
##                               0                               0
##                gender.nonbinary               family_history.No
##                               0                               0
##              family_history.Yes             benefits.Don't know
##                               0                               0
##                     benefits.No                    benefits.Yes
##                               0                               0
##            anonymity.Don't know                    anonymity.No
##                               0                               0
##                   anonymity.Yes                   supervisor.No
##                               0                               0
##         supervisor.Some of them                  supervisor.Yes
##                               0                               0
##                    coworkers.No          coworkers.Some of them
##                               0                               0
##                   coworkers.Yes mental_health_consequence.Maybe
##                               0                               0
##    mental_health_consequence.No   mental_health_consequence.Yes
##                               0                               0
##   phys_health_consequence.Maybe      phys_health_consequence.No
##                               0                               0
##     phys_health_consequence.Yes   mental_vs_physical.Don't know
##                               0                               0
##           mental_vs_physical.No          mental_vs_physical.Yes
##                               0                               0
##                leave.Don't know        leave.Somewhat difficult
##                               0                               0
##             leave.Somewhat easy            leave.Very difficult
##                               0                               0
##                 leave.Very easy              reason_for_absence
##                               0                               0
##                month_of_absence distance_from_residence_to_work
##                               0                               0
##                    service_time                  social_drinker
##                               0                               0
##                   social_smoker            disciplinary_failure
##                               0                               0
##                 body_mass_index       absenteeism_time_in_hours
##                               0                               0
##                             pet                             son
##                                0                               0
##                              age
##                                0
```

```
rf_model <- randomForest(
  x = X_train,
  y = as.factor(y_train),
  ntree = 600,
  mtry = floor(sqrt(ncol(X_train))),
  importance = TRUE
)

pred <- predict(rf_model, X_test)
cm <- confusionMatrix(pred, as.factor(y_test))
cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 230  66
##          1  30 107
##
##                Accuracy : 0.7783
##                  95% CI : (0.7362, 0.8166)
##     No Information Rate : 0.6005
##     P-Value [Acc > NIR] : 3.102e-15
##
##                   Kappa : 0.5213
##
##  Mcnemar's Test P-Value : 0.000354
##
##             Sensitivity : 0.8846
##             Specificity : 0.6185
##          Pos Pred Value : 0.7770
##          Neg Pred Value : 0.7810
##              Prevalence : 0.6005
##          Detection Rate : 0.5312
##    Detection Prevalence : 0.6836
##       Balanced Accuracy : 0.7516
##
##        'Positive' Class : 0
##
```

```
metricas <- tibble(
  accuracy = cm$overall["Accuracy"],
  sensitivity = cm$byClass["Sensitivity"],
  specificity = cm$byClass["Specificity"],
  precision = cm$byClass["Precision"],
  recall = cm$byClass["Recall"],
  f1 = cm$byClass["F1"]
)

metricas

## # A tibble: 1 × 6
##   accuracy sensitivity specificity precision recall    f1
##      <dbl>       <dbl>       <dbl>     <dbl>  <dbl> <dbl>
## 1    0.778       0.885       0.618     0.777  0.885 0.827

varImpPlot(rf_model)
```

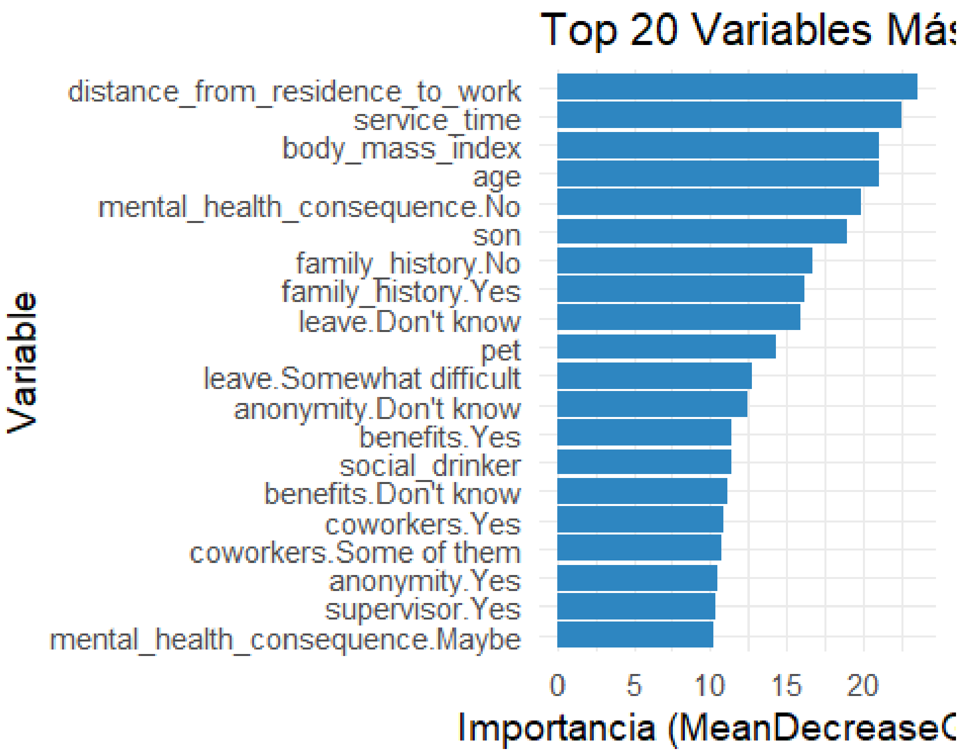


```
importance(rf_model)

##                                             0          1
MeanDecreaseAccuracy
## gender.female                        4.530564  5.1593224
6.44268305
## gender.male                          8.477285  7.2123990
10.36107831
## gender.nonbinary                     1.831966 -1.7320901          -
0.01398113
## family_history.No                   16.644717 15.3752276
18.36708252
## family_history.Yes                  16.134144 15.4959123
17.99114448
## benefits.Don't know                 11.176606  2.7450575
10.12891883
```

```
## benefits.No                      7.947762  6.8568211
9.03025780
## benefits.Yes                    11.404686  8.9886295
11.99766398
## anonymity.Don't know            12.496204 10.3803153
13.50352772
## anonymity.No                     0.408588  2.3620290
2.19639269
## anonymity.Yes                   10.456398  3.8478059
9.61834803
## supervisor.No                    8.157145 -1.2154775
6.27928954
## supervisor.Some of them         10.035369  3.6150127
9.53671320
## supervisor.Yes                  10.387993  4.3208858
10.44897904
## coworkers.No                     3.403638  1.9493088
3.83077456
## coworkers.Some of them          10.783506 10.1827510
11.69211919
## coworkers.Yes                   10.928174  8.6634076
11.33105184
## mental_health_consequence.Maybe 10.208355  8.3066757
10.69799244
## mental_health_consequence.No    19.924532 16.0317549
21.25032406
## mental_health_consequence.Yes    6.278494  9.5618306
9.89927189
## phys_health_consequence.Maybe    4.732769 -2.6828432
2.18582310
## phys_health_consequence.No       6.839304 -2.6173822
4.44686265
## phys_health_consequence.Yes      5.674073  3.9398288
6.29243781
## mental_vs_physical.Don't know    7.395002  5.8439795
8.38453306
## mental_vs_physical.No           10.074183 13.0793485
14.07651954
## mental_vs_physical.Yes           9.632156  5.1972636
9.83622840
## leave.Don't know                15.898478 10.5151923
16.78651865
## leave.Somewhat difficult        12.710014  1.7186352
12.67066055
## leave.Somewhat easy              2.375696  3.2262519
3.56902563
## leave.Very difficult             4.951809  8.7392048
9.72458384
## leave.Very easy                  9.382742  3.4434304
8.07701072
```

```
## reason_for_absence               1.688187  4.2254618
5.71616533
## month_of_absence                 3.281581 -0.9306166
2.68309412
## distance_from_residence_to_work 23.620270 19.4019156
26.04438496
## service_time                    22.475484 19.3598016
25.07548459
## social_drinker                  11.374805 11.1927080
14.25917972
## social_smoker                    8.192077  9.3331908
10.53235653
## disciplinary_failure             2.865266 -2.3928685
0.44191968
## body_mass_index                 21.079285 19.0491289
23.72565053
## absenteeism_time_in_hours        3.081859  2.2836148
4.90858996
## pet                             14.359471 15.3042100
17.65746223
## son                             18.941599 17.6798319
20.74631610
## age                             21.014824 17.8709906
23.41377458
##                                 MeanDecreaseGini
## gender.female                           3.325145
## gender.male                             5.185991
## gender.nonbinary                        1.391392
## family_history.No                      15.310924
## family_history.Yes                     15.290003
## benefits.Don't know                     6.876282
## benefits.No                             7.431069
## benefits.Yes                           10.172130
## anonymity.Don't know                   10.469059
## anonymity.No                            2.382246
## anonymity.Yes                           6.689565
## supervisor.No                           5.539580
## supervisor.Some of them                 7.068121
## supervisor.Yes                          7.225160
## coworkers.No                            5.266410
## coworkers.Some of them                 10.068294
## coworkers.Yes                           7.859204
## mental_health_consequence.Maybe         8.545483
## mental_health_consequence.No           25.771546
## mental_health_consequence.Yes           7.387605
## phys_health_consequence.Maybe           4.780339
## phys_health_consequence.No              5.814788
## phys_health_consequence.Yes             3.309244
## mental_vs_physical.Don't know           7.708926
## mental_vs_physical.No                  10.566163
## mental_vs_physical.Yes                     6.966137
## leave.Don't know                          13.041848
## leave.Somewhat difficult                   5.840741
## leave.Somewhat easy                        5.705039
## leave.Very difficult                       5.883753
## leave.Very easy                            5.873235
## reason_for_absence                         3.985620
## month_of_absence                           3.116818
## distance_from_residence_to_work           29.922063
## service_time                              24.217771
## social_drinker                             6.881576
## social_smoker                              2.960804
## disciplinary_failure                       0.208078
## body_mass_index                           22.762686
## absenteeism_time_in_hours                  3.720002
## pet                                        9.810564
## son                                       17.761998
## age                                       21.783498
```

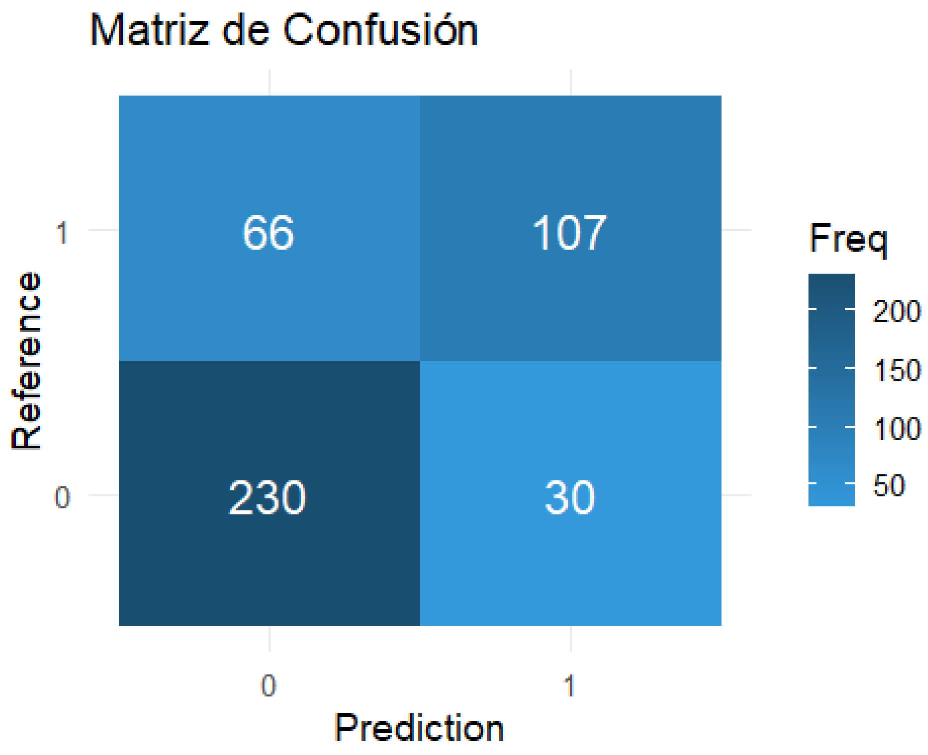
```
library(ggplot2)

# Obtener importancia
imp <- importance(rf_model)
imp_df <- data.frame(
  Variable = rownames(imp),
  Importance = imp[, 1]
)

# Ordenar
imp_df <- imp_df %>% arrange(desc(Importance))

# Plot
ggplot(imp_df[1:20, ], aes(x = reorder(Variable, Importance), y =
Importance)) +
  geom_col(fill = "#2E86C1") +
  coord_flip() +
  labs(
    title = "Top 20 Variables Más Importantes",
    x = "Variable",
    y = "Importancia (MeanDecreaseGini)"
  ) +
  theme_minimal(base_size = 14)
```



```
library(dplyr)

tabla_importancia <- imp_df %>%
  arrange(desc(Importance)) %>%
  head(20)

tabla_importancia

##                                                         Variable
Importance
## distance_from_residence_to_work distance_from_residence_to_work
23.62027
## service_time                                       service_time
22.47548
## body_mass_index                                 body_mass_index
21.07929
## age                                                         age
21.01482
## mental_health_consequence.No       mental_health_consequence.No
19.92453
## son                                                         son
18.94160
## family_history.No                             family_history.No
16.64472
## family_history.Yes                           family_history.Yes
16.13414
## leave.Don't know                               leave.Don't know
```

```
15.89848
## pet                                                         pet
14.35947
## leave.Somewhat difficult               leave.Somewhat difficult
12.71001
## anonymity.Don't know                       anonymity.Don't know
12.49620
## benefits.Yes                                       benefits.Yes
11.40469
## social_drinker                                   social_drinker
11.37480
## benefits.Don't know                         benefits.Don't know
11.17661
## coworkers.Yes                                     coworkers.Yes
10.92817
## coworkers.Some of them                   coworkers.Some of them
10.78351
## anonymity.Yes                                     anonymity.Yes
10.45640
## supervisor.Yes                                   supervisor.Yes
10.38799
## mental_health_consequence.Maybe mental_health_consequence.Maybe
10.20835
```

```r
library(caret)
library(ggplot2)

cm_table <- as.data.frame(cm$table)

ggplot(cm_table, aes(Prediction, Reference, fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), color = "white", size = 6) +
  scale_fill_gradient(low = "#3498DB", high = "#1B4F72") +
  labs(title = "Matriz de Confusión") +
  theme_minimal(base_size = 14)
```



Shiny

```
library(shiny)

## Warning: package 'shiny' was built under R version 4.5.2

library(tidyverse)
library(ggplot2)
library(caret)

# Aquí asumimos que ya tienes en memoria:
# - rf_model
# - cm
# - metricas
# - imp_df (importancia de variables)
# - predictoras (dataset con las features + risk)

ui <- fluidPage(
  titlePanel("Dashboard de Riesgo de Salud Mental"),

  sidebarLayout(
    sidebarPanel(
      h4("Controles"),
      selectInput(
        "var_color",
        "Color por variable categórica:",
        choices = c("gender", "family_history", "benefits", "leave"),
        selected = "gender"
      ),
      hr(),
      h4("Predicción individual"),
      selectInput("inp_gender", "Género:",
                  choices = c("male", "female", "nonbinary")),
      selectInput("inp_family_history", "Antecedentes familiares:",
                  choices = c("Yes", "No")),
      selectInput("inp_benefits", "Beneficios de salud mental:",
                  choices = c("Yes", "No", "Don't know")),
      selectInput("inp_leave", "Facilidad para pedir baja:",
                  choices = c("Very easy", "Somewhat easy", "Somewhat
difficult",
                              "Very difficult", "Don't know")),
      numericInput("inp_bmi", "Índice de masa corporal (BMI):", 25, 10,
50, 0.5),
      numericInput("inp_distance", "Distancia casa-trabajo (km):", 10, 0,
100, 1),
      actionButton("btn_predict", "Predecir riesgo")
    ),

    mainPanel(
      tabsetPanel(
        tabPanel("Resumen",
                 h3("Métricas del modelo"),
                 tableOutput("tbl_metricas"),
                 h4("Matriz de confusión"),
                 plotOutput("plot_cm")
        ),
        tabPanel("Importancia de variables",
                 h3("Top 20 variables más importantes"),
                 plotOutput("plot_importancia"),
                 tableOutput("tbl_importancia")
        ),
        tabPanel("Distribuciones",
                 h3("Riesgo por variable seleccionada"),
                 plotOutput("plot_riesgo_var")
        ),
        tabPanel("Predicción individual",
                 h3("Resultado de la predicción"),
                 verbatimTextOutput("txt_prediccion")
        )
      )
    )
  )
)

server <- function(input, output, session) {
```

```r
  # ---- Tabla de métricas ----
  output$tbl_metricas <- renderTable({
    metricas
  }, rownames = TRUE)

  # ---- Matriz de confusión visual ----
  output$plot_cm <- renderPlot({
    cm_table <- as.data.frame(cm$table)

    ggplot(cm_table, aes(Prediction, Reference, fill = Freq)) +
      geom_tile() +
      geom_text(aes(label = Freq), color = "white", size = 6) +
      scale_fill_gradient(low = "#3498DB", high = "#1B4F72") +
      labs(title = "Matriz de Confusión") +
      theme_minimal(base_size = 14)
  })

  # ---- Importancia de variables (plot) ----
  output$plot_importancia <- renderPlot({
    ggplot(imp_df[1:20, ], aes(x = reorder(Variable, Importance), y =
Importance)) +
      geom_col(fill = "#2E86C1") +
      coord_flip() +
      labs(
        title = "Top 20 Variables Más Importantes",
        x = "Variable",
        y = "Importancia (MeanDecreaseGini)"
      ) +
      theme_minimal(base_size = 14)
  })

  # ---- Importancia de variables (tabla) ----
  output$tbl_importancia <- renderTable({
    imp_df[1:20, ]
  }, rownames = FALSE)

  # ---- Distribución de riesgo por variable categórica ----
  output$plot_riesgo_var <- renderPlot({
    var_sel <- sym(input$var_color)

    predictoras %>%
      mutate(risk = factor(risk, labels = c("No riesgo", "En riesgo")))
%>%
      ggplot(aes(x = !!var_sel, fill = risk)) +
      geom_bar(position = "fill") +
      scale_y_continuous(labels = scales::percent) +
      labs(
        title = paste("Proporción de riesgo por", input$var_color),
        x = input$var_color,
        y = "% dentro de cada categoría",
        fill = "Riesgo"
      ) +
      theme_minimal(base_size = 14) +
      theme(axis.text.x = element_text(angle = 45, hjust = 1))
  })

  # ---- Predicción individual ----
  observeEvent(input$btn_predict, {
    # Construir un data.frame con una sola fila
    new_data <- tibble(
      gender = factor(input$inp_gender, levels = c("male", "female",
"nonbinary")),
      family_history = factor(input$inp_family_history, levels = c("Yes",
"No")),
      benefits = factor(input$inp_benefits, levels = c("Yes", "No",
"Don't know")),
      anonymity = factor("Yes", levels = c("Yes", "No", "Don't know")),
      supervisor = factor("Yes", levels = c("Yes", "No", "Some of
them")),
      coworkers = factor("Yes", levels = c("Yes", "No", "Some of them")),
      mental_health_consequence = factor("Maybe", levels = c("Yes", "No",
"Maybe")),
      phys_health_consequence = factor("Maybe", levels = c("Yes", "No",
"Maybe")),
      mental_vs_physical = factor("Don't know", levels = c("Yes", "No",
"Don't know")),
      leave = factor(input$inp_leave,
                     levels = c("Very easy", "Somewhat easy", "Somewhat
difficult",
                                "Very difficult", "Don't know")),
      reason_for_absence = 1,
      month_of_absence = 1,
      distance_from_residence_to_work =
        as.numeric(scale(input$inp_distance,
                         center =
attr(scale(predictoras$distance_from_residence_to_work),
"scaled:center"),
                         scale  =
attr(scale(predictoras$distance_from_residence_to_work),
"scaled:scale"))),
      service_time = 0,
      social_drinker = 0,
      social_smoker = 0,
      disciplinary_failure = 0,
      body_mass_index =
        as.numeric(scale(input$inp_bmi,
                         center =
attr(scale(predictoras$body_mass_index), "scaled:center"),
                         scale  =
attr(scale(predictoras$body_mass_index), "scaled:scale"))),
      absenteeism_time_in_hours = 0,
      pet = 0,
      son = 0,
      age =
        as.numeric(scale(30,
                         center = attr(scale(predictoras$age),
"scaled:center"),
                         scale  = attr(scale(predictoras$age),
"scaled:scale")))
    )

    # Aplicar mismas dummies que al entrenamiento
    dummies_new <- dummyVars(~ ., data = predictoras %>% select(-risk))
    X_new <- predict(dummies_new, newdata = new_data) %>% as.data.frame()

    # Predicción
    pred_new <- predict(rf_model, X_new, type = "prob")

    riesgo <- round(pred_new[,"1"] * 100, 1)

    output$txt_prediccion <- renderText({
      paste0("Probabilidad estimada de estar EN RIESGO: ", riesgo, "%")
    })
  })
}

shinyApp(ui = ui, server = server)
```

Ejemplo práctico de la ejecución del modelo predictivo

```
set.seed(123)

n <- 500  # número de empleados ficticios

empleados <- tibble(
  id = 1:n,
  nombre = paste0("Empleado_", 1:n),
  departamento = sample(
    c("Operaciones", "Ventas", "IT", "RRHH", "Finanzas"),
    n, replace = TRUE
  ),
  puesto = sample(
    c("Analista", "Coordinador", "Gerente", "Ejecutivo", "Especialista"),
    n, replace = TRUE
  ),
  productividad = round(runif(n, 60, 100), 1),
  carga_laboral = round(runif(n, 70, 130), 1)
)
```

```r
muestras <- predictoras[sample(1:nrow(predictoras), n, replace = TRUE), ]

# Aplicar las mismas dummies que en el entrenamiento
dummies_pred <- dummyVars(~ ., data = predictoras %>% select(-risk))
X_sintetico <- predict(dummies_pred, newdata = muestras) %>%
as.data.frame()

# Predicción de riesgo
pred_riesgo <- predict(rf_model, X_sintetico, type = "prob")

empleados$riesgo <- pred_riesgo[, "1"]
empleados$riesgo_nivel <- case_when(
  empleados$riesgo >= 0.66 ~ "Alto",
  empleados$riesgo >= 0.33 ~ "Medio",
  TRUE ~ "Bajo"
)

kpi_riesgo_promedio <- mean(empleados$riesgo)
kpi_riesgo_alto <- mean(empleados$riesgo_nivel == "Alto")
kpi_productividad <- mean(empleados$productividad)
kpi_carga <- mean(empleados$carga_laboral)

library(bslib)
```

```
##
## Adjuntando el paquete: 'bslib'

## The following object is masked from 'package:utils':
##
##     page
```

```r
library(shinyWidgets)
```

```
## Warning: package 'shinyWidgets' was built under R version 4.5.2
```

```r
ui <- fluidPage(
  theme = bs_theme(
    version = 5,
    bootswatch = "flatly",
    primary = "#2E86C1",
    secondary = "#1B4F72",
    base_font = font_google("Inter")
  ),

  navbarPage(
    title = div(
      style = "font-weight:700; font-size:22px; color:#1B4F72;",
      "Impulso — Bienestar y Riesgo"
    ),

    # ---------------- DASHBOARD GENERAL ----------------
    tabPanel("Dashboard General",
      br(),
      fluidRow(
        column(3,
          card(
            card_header("Nivel de Riesgo"),
            h2(style="color:#C0392B;",
paste0(round(kpi_riesgo_promedio*100,1), "%")),
            p("Promedio general")
          )
        ),
        column(3,
          card(
            card_header("Riesgo Alto"),
            h2(style="color:#C0392B;",
paste0(round(kpi_riesgo_alto*100,1), "%")),
            p("Empleados en riesgo alto")
          )
        ),
        column(3,
          card(
            card_header("Productividad"),
            h2(style="color:#2E86C1;", paste0(round(kpi_productividad,1),
"%")),
            p("Último mes")
          )
        ),
        column(3,
          card(
            card_header("Carga Laboral"),
            h2(style="color:#1B4F72;", paste0(round(kpi_carga,1), "%")),
            p("Promedio general")
          )
        )
      ),
      br(),
      card(
        card_header("Tendencia de Rendimiento y Riesgo"),
        plotOutput("plot_tendencia", height = "300px")
      ),
      br(),
      card(
        card_header("Comparación por Área"),
        tableOutput("tabla_comparacion")
      )
    ),

    # ---------------- EMPLEADOS ----------------
    tabPanel("Empleados",
      sidebarLayout(
        sidebarPanel(
          selectInput("filtro_dep", "Departamento:",
                      choices = unique(empleados$departamento)),
          textInput("buscar", "Buscar empleado:")
        ),
        mainPanel(
          card(
            card_header("Listado de Empleados"),
            tableOutput("tabla_empleados")
          )
        )
      )
    ),

    # ---------------- ANALISIS DE RIESGO ----------------
    tabPanel("Análisis de Riesgo",
      card(
        card_header("Riesgo por Departamento"),
        plotOutput("plot_riesgo_dep")
      ),
      br(),
      card(
        card_header("Distribución del Riesgo"),
        plotOutput("plot_riesgo_hist")
      )
    ),

    # ---------------- FACTORES DE RIESGO ----------------
    tabPanel("Factores de Riesgo",
      card(
        card_header("Importancia de Variables"),
        plotOutput("plot_importancia")
      )
    ),

    # ---------------- SEGMENTACIÓN (CORREGIDA) ----------------
    tabPanel("Segmentación",
      br(),
      card(
        card_header("Resumen por Departamento"),
        tableOutput("tabla_segmentacion")
      ),
      br(),
      fluidRow(
        column(6,
          card(
            card_header("Riesgo Promedio por Departamento"),
            plotOutput("plot_seg_riesgo")
          )
        ),
        column(6,
          card(
            card_header("Productividad Promedio por Departamento"),
            plotOutput("plot_seg_prod")
          )
        )
      ),
      br(),
      card(
        card_header("Distribución de Niveles de Riesgo por
Departamento"),
        plotOutput("plot_seg_niveles")
      )
    ),

    # ---------------- RECOMENDACIONES ----------------
    tabPanel("Recomendaciones",
      card(
        card_header("Sugerencias Automáticas"),
        verbatimTextOutput("txt_recomendaciones")
      )
    )
  )
)

server <- function(input, output, session) {

  # ---------------- TENDENCIA (CORREGIDA) ----------------
  output$plot_tendencia <- renderPlot({
    df <- tibble(
      mes = factor(c("Ene", "Feb", "Mar", "Abr", "May", "Jun"),
                   levels = c("Ene", "Feb", "Mar", "Abr", "May", "Jun")),
      rendimiento = seq(75, 85, length.out = 6),
      riesgo = seq(25, 35, length.out = 6)
    )

    ggplot(df, aes(mes)) +
      geom_line(aes(y = rendimiento, color = "Rendimiento"), size = 1.5)
+
      geom_point(aes(y = rendimiento, color = "Rendimiento"), size = 3) +
      geom_line(aes(y = riesgo, color = "Riesgo"), size = 1.5) +
      geom_point(aes(y = riesgo, color = "Riesgo"), size = 3) +
      scale_color_manual(values = c("Rendimiento" = "#2E86C1", "Riesgo" =
"#C0392B")) +
      theme_minimal(base_size = 16) +
      labs(title = "Tendencia de Rendimiento y Riesgo", y = "Valor", x =
"Mes", color = "")
  })

  # ---------------- TABLA COMPARACIÓN ----------------
  output$tabla_comparacion <- renderTable({
    empleados %>%
      group_by(departamento) %>%
      summarise(
        empleados = n(),
        riesgo_promedio = round(mean(riesgo) * 100, 1),
        productividad_promedio = round(mean(productividad), 1),
        carga_promedio = round(mean(carga_laboral), 1)
      )
  })

  # ---------------- EMPLEADOS ----------------
  output$tabla_empleados <- renderTable({
    empleados %>%
      filter(departamento == input$filtro_dep) %>%
      filter(str_detect(nombre, regex(input$buscar, ignore_case = TRUE)))
  })

  # ---------------- RIESGO POR DEP ----------------
  output$plot_riesgo_dep <- renderPlot({
    empleados %>%
      group_by(departamento) %>%
      summarise(riesgo_promedio = mean(riesgo)) %>%
      ggplot(aes(x = reorder(departamento, riesgo_promedio),
                 y = riesgo_promedio,
                 fill = departamento)) +
      geom_col() +
      coord_flip() +
      theme_minimal(base_size = 14) +
      labs(y = "Riesgo Promedio", x = "Departamento") +
      scale_fill_brewer(palette = "Set2")
  })

  # ---------------- HISTOGRAMA RIESGO ----------------
  output$plot_riesgo_hist <- renderPlot({
    ggplot(empleados, aes(riesgo)) +
      geom_histogram(fill = "#2E86C1", bins = 20) +
      theme_minimal(base_size = 14) +
      labs(title = "Distribución del riesgo", x = "Riesgo")
  })

  # ---------------- IMPORTANCIA ----------------
  output$plot_importancia <- renderPlot({
    ggplot(imp_df[1:20, ], aes(x = reorder(Variable, Importance), y =
Importance)) +
      geom_col(fill = "#C0392B") +
      coord_flip() +
      theme_minimal(base_size = 14) +
      labs(title = "Top 20 factores de riesgo", x = "Variable", y =
"Importancia")
  })

  # ---------------- SEGMENTACIÓN (CORREGIDA) ----------------

  output$tabla_segmentacion <- renderTable({
    empleados %>%
      group_by(departamento) %>%
      summarise(
        empleados = n(),
        riesgo_promedio = round(mean(riesgo) * 100, 1),
        productividad_promedio = round(mean(productividad), 1),
        carga_promedio = round(mean(carga_laboral), 1)
      )
  })

  output$plot_seg_riesgo <- renderPlot({
    empleados %>%
      group_by(departamento) %>%
      summarise(riesgo_promedio = mean(riesgo)) %>%
      ggplot(aes(x = reorder(departamento, riesgo_promedio),
                 y = riesgo_promedio,
                 fill = departamento)) +
      geom_col() +
      coord_flip() +
      theme_minimal(base_size = 14) +
      labs(y = "Riesgo Promedio", x = "Departamento") +
      scale_fill_brewer(palette = "Set2")
  })

  output$plot_seg_prod <- renderPlot({
    empleados %>%
      group_by(departamento) %>%
      summarise(productividad_promedio = mean(productividad)) %>%
      ggplot(aes(x = reorder(departamento, productividad_promedio),
                 y = productividad_promedio,
                 fill = departamento)) +
      geom_col() +
      coord_flip() +
      theme_minimal(base_size = 14) +
      labs(y = "Productividad Promedio", x = "Departamento") +
      scale_fill_brewer(palette = "Set3")
  })

  output$plot_seg_niveles <- renderPlot({
    empleados %>%
      ggplot(aes(x = departamento, fill = riesgo_nivel)) +
      geom_bar(position = "fill") +
      scale_y_continuous(labels = scales::percent) +
```

```r
      theme_minimal(base_size = 14) +
      labs(y = "% dentro del departamento", x = "Departamento", fill =
"Nivel de Riesgo") +
      scale_fill_manual(values = c("Bajo" = "#2ECC71", "Medio" =
"#F1C40F", "Alto" = "#E74C3C"))
  })

  # ---------------- RECOMENDACIONES ----------------
  output$txt_recomendaciones <- renderText({
    paste(
      "- Reducir carga laboral en Operaciones.",
      "- Implementar pausas activas en áreas con riesgo alto.",
      "- Revisar políticas de apoyo psicológico.",
      "- Monitorear empleados con riesgo > 0.7.",
      sep = "\n"
    )
  })
}

shinyApp(ui = ui, server = server)
```