
```

1
2 #
   -----

3 #           PASOS PARA CREAR UN PAQUETE EN R STUDIOS
4 #
   -----

5
6
7 #
   -----

8 # 1) Instalar los paquetes devtools y usethis, para luego llamar a
   las librerías
9 #
   -----

10
11 install.packages("devtools")
12 install.packages("usethis")
13
14 library(usethis)
15 library(devtools)
16
17
18 #
   -----

19 # 2) Creamos el nuevo paquete usando la función "create_package()",
   especifican-
20 # do la dirección junto con el nombre del paquete con el que se
   desea crear
21 # el proyecto.
22 #
   -----

23
24 usethis::create_package("C:/Users/Usuario01/Documents/Pasantes/
   Jonathan_Pallasco
25                               /Nuevo")
26
27 # Se nos crea un nuevo proyecto con 6 archivos de los cuales se
   nombran a
28 # continuación
29 # 1) .Rbuildignore
30 # 2) .gitignore
31 # 3) DESCRIPTION
32 # 4) NAMESPACE
33 # 5) Directorio R
34 # 6) Nuevo.Rproj
35
36
37 #
   -----

```

```

38 # 3) El directorio creado ("Nuevo"), es un paquete de R y un
    proyecto RStudio.
39 #   Ahora se debe convertir en un repositorio Git, usando la
    funcion use git().
40 #   Cabe recalcar que, use git() funciona en cualquier proyecto,
41 #   independientemente de si es un paquete R.
42 #
    -----

43
44 usethis::use_git()
45
46 #
    -----

47 # 4) Escribir la primera función: Toda funcion esta en un archivo .
    R
48 #   (Nombre_Funcion.R), y se va a encontrar almacenada en la
    carpeta R/ del
49 #   proyecto. La funcion use_r() del paquete usethis, crea y/o
    abre un script
50 #   en la carpeta R con el nombre que le asignemos.
51 #
    -----

52
53 # Creamos funciones
54 usethis::use_r('Suma')
55 usethis::use_r('Sumatoria')
56 usethis::use_r('Media')
57
58
59 # Se nos abren scrips en la carpeta R, llamado Suma.R, Sumatoria.R,
    Media.R y
60 # empezamos a crear funciones simples
61
62 funcion_suma <- function(a,b){
63   suma <- a + b
64   return(suma)
65 }
66
67 Sumatoria <- function(vector){
68   n <- length(vector)
69   sumatoria <- 0
70   for (i in 1:n){
71     sumatoria <- sumatoria + vector[i]
72   }
73   return(sumatoria)
74 }
75
76 Media <- function(vector){
77   n <- length(vector)
78   media <- Sumatoria(vector)/n
79   return(media)
80 }

```

```

81
82
83 # -----

84 # 5) Una vez, creada una funcion simple, debemos llamar a todas las
      funciones,
85 #     a disposición la experimentación, utilizando la funcion "load_
      all('.')"
86 #     del paquete devtools
87 # -----

88
89 devtools::load_all('.')
90
91 # Comprobación del funcionamiento de la funcion "funcion_suma"
92
93 a <- 10
94 b <- 21.31
95 vector <- c(5, 2, 4, 5.3, 10.23, 15, 1.23)
96 Suma(a,b)
97 Sumatoria(vector)
98 Media(vector)
99
100 # Verificar si las funciones existen en el entorno global
101 exists("Suma", where = globalenv(), inherits = FALSE)
102 exists("Sumatoria", where = globalenv(), inherits = FALSE)
103 exists("Media", where = globalenv(), inherits = FALSE)
104
105
106 # Tenemos evidencia empírica informal, que las funciones, funciona
107 # correctamente. Pero para asegurar que todas las partes móviles
      del nuevo
108 # paquete funcionan?.
109
110 # -----

111 # 6) Usar la función check() del paquete devtools es una forma
      conveniente de
112 #     verificar que las partes móviles funcione sin salir de la
      sección R.
113 # -----

114
115 devtools::check()
116
117 # -----

118 # 7) Creamos nuestra primera base de datos de la información acerca
      de las
119 #     familias y guardamos en un directorio llamado data, usando la
      funcion

```

```

120 # create_data()
121 #
-----

122
123 install.packages("data.table")
124 library(data.table)
125 Familia <- as.data.table( c(1:200) )
126 setnames(Familia, c('Id'))
127 Familia <- Familia[ , Personas := sample(1:10, size = 200,
128                                           replace = TRUE) ]
129 Familia[ Personas <= 4, Tipo := 'Familia Normal' ]
130 Familia[ Personas >= 4 & Personas <= 6 ,
131         Tipo := 'Familia Media' ]
132 Familia[ Personas >= 7, Tipo := 'Familia Numerosa' ]
133
134 usethis::create_data(Familia)
135
136 #
-----

137 # 8) Prueba de que las funciones y base da datos funcionan
    correctamente
138 #
-----

139
140 devtools::load_all('.')
141 data(Familia)
142 Sumatoria(Familia$Personas)
143 Media(Familia$Personas)
144
145 #
-----

146 # 9) Metadatos del paquete
147 #
-----

148
149
150 #
-----

151 # 9.1) DESCRIPTION
152 #
-----

153
154 # Titulo
155 Title: Nuevo Paquete R de Funciones Simples
156 # Descripcion
157 Description: Utilizar funciones simples a menudo se complican
158             demasida dada esta circunstancia se emplean 3 funciones:
159             'Suma', 'Sumatoria', 'Media' para realizar calculos rapidos
160             sin la necesidad de escribir varias lineas de codigo.

```

```

161 # Unico autor
162 Authors@R:
163   person("Jonathan", "Pallasco",
164         email = "jonathan.pallasco@epn.edu.ec",
165         role = c("aut", "cre"),
166         comment = c(ORCID = "0000-0003-2524-9390")
167
168 # Varios autores
169 Authors@R:c(
170   person("Jonathan", "Pallasco",
171         email = "jonathan.pallasco@epn.edu.ec",
172         role = c("aut", "cre"),
173         comment = c(ORCID = "0000-0003-2524-9390"),
174   person("Miguel", "Flores",
175         email = "jonathan.pallasco@epn.edu.ec",
176         role = c("aut", "cre"),
177         comment = c(ORCID = "0000-0002-7742-1247"))
178
179 # Url y BugReports
180 usethis::use_github()
181
182 #
-----
183 # 9.2) NAMESPACE
184 #
-----
185
186 install.packages(roxigen2)
187 library(roxigen2)
188 # Nota: No recomiendo escribir estas directivas a mano.
189
190 #
-----
191 # 10) Documentación de las funciones y bases de datos
192 #
-----
193
194 #' Add together two numbers
195 #'
196 #' @param x A number.
197 #' @param y A number.
198 #' @return A numeric vector.
199 #' @examples
200 #' add(1, 1)
201 #' add(10, 1)
202 add <- function(x, y) {
203   x + y
204 }

```
