



**Escuela Universitaria  
Politécnica** - La Almunia  
Centro adscrito  
**Universidad Zaragoza**

**ESCUELA UNIVERSITARIA POLITÉCNICA  
DE LA ALMUNIA DE DOÑA GODINA (ZARAGOZA)**

**ANEXOS**

Optimización de red de distribución de  
alimentos

Food distribution network optimization

425.20.44

Autor: Raquel Escalera Lapuerta

Director: Luis Mariano Esteban

Fecha: 22/06/2020



## INDICE DE CONTENIDO

<b>ANEXO 1. CÓDIGO COMPLETO EMPLEADO</b>	<b>1</b>
1.1. CÓDIGO PARA OBTENCIÓN DE LA SOLUCIÓN	1
1.2. CÓDIGO PARA LA APLICACIÓN	3
<b>ANEXO 2. MANUAL DE INSTALACIÓN DE LA APP</b>	<b>10</b>
<b>ANEXO 3. MATRIZ DE DISTANCIAS</b>	<b>12</b>
<b>ANEXO 4. TABLA DE COORDENADAS DE MUNICIPIOS</b>	<b>13</b>

## INDICE DE ILUSTRACIONES

Ilustración 1. Cargar archivos desde RStudio.....	10
Ilustración 2. Correr aplicación. ....	11

## INDICE DE TABLAS

Tabla 1. Matriz de distancias. ....	12
Tabla 2. Listado de coordenadas. ....	13

## ANEXO 1. CÓDIGO COMPLETO EMPLEADO

### 1.1. CÓDIGO PARA OBTENCIÓN DE LA SOLUCIÓN

```
library(readxl)

library(leaflet)

library(TSP)

#devtools::install_github("GIScience/openrouteservice-r")

library(openrouteservice)

#Cargo los datos de Excel

setwd("C:/Directorio de trabajo")

MatrizDistancias<-read_excel("MatrizDistancias.xlsx",col_names=TRUE)

Coordenadas<-read_excel("Coordenadas.xlsx",col_names = TRUE)

#Convierto los datos cargados en matrices y establezco el rownames correspondiente a cada matriz

MatrizTrabajo<-as.matrix(MatrizDistancias)

rownames(MatrizTrabajo)<-colnames(MatrizTrabajo)

Coordenadas<-as.matrix(Coordenadas)

coordenadas<-Coordenadas

ID1=rownames(coordenadas)<-rownames(MatrizTrabajo)

#Establezco la matriz de coordenadas como un data frame para poder representar la ubicación de los municipios en el mapa

coordenadas<-as.data.frame(coordenadas)

mapa <- leaflet(data = coordenadas) %>%

  addTiles() %>%
```

## Código completo empleado

```
setView(Ing = coordenadas$LONGITUD[9], lat = coordenadas$LATITUD[9] ,  
zoom = 9) %>%
```

```
addMarkers(~LONGITUD, ~LATITUD, popup = ~as.character(ID1), label =  
~as.character(ID1))
```

```
mapa
```

**#Obtención de la ruta más optima**

```
MatrizTSP<-TSP(as.matrix(MatrizTrabajo))
```

```
Solucion1<-
```

```
solve_TSP((MatrizTSP),method="nearest_insertion",control=list(start=1))
```

```
Solucion2<-
```

```
solve_TSP((MatrizTSP),method="farthest_insertion",control=list(start=1))
```

```
Longitud1<-tour_length(Solucion1)
```

```
Longitud2<-tour_length(Solucion2)
```

```
if(Longitud1<Longitud2){
```

```
  SolucionRuta=Solucion1
```

```
}else{
```

```
  SolucionRuta=Solucion2
```

```
}
```

```
SolucionRuta
```

**#Listado en municipios en el orden que deben visitarse**

```
Ruta<-data.frame(SolucionRuta)
```

```
Ruta
```

**#Construcción de la ruta en el orden que debe visitarse para  
representación en el mapa**

```
coordenadasok<-coordenadas[attributes(SolucionRuta)$names,]
```

```
CoordenadasFase<-cbind(coordenadasok[,1],coordenadasok[,2])
CoordenadasFase<-rbind(CoordenadasFase,CoordenadasFase[1,])

coordenadasFasef<-function(x){CoordenadasFase[x,]}
coordenadasFase<-lapply(seq(1,dim(CoordenadasFase)[1]),coordenadasFasef)

ors_api_key('5b3ce3597851110001cf62485fce866458364a43b3aa39e829eef59c'
)

trayecto<-
ors_directions(coordenadasFase,format="geojson",radiuses=rep(10000,length(coorde
nadasFase)))

ID1=rownames(coordenadasok)<-rownames(coordenadasok)

leaflet(data = coordenadasok) %>%
  addTiles() %>%
  addMarkers(~LONGITUD, ~LATITUD, popup = ~as.character(ID1), label =
~as.character(ID1))%>%
  addGeoJSON(trayecto, fill=FALSE) %>%
  fitBBox(trayecto$bbox)
```

## 1.2. CÓDIGO PARA LA APLICACIÓN

```
library(shiny)
library(TSP)
library(leaflet)
library(readxl)
#devtools::install_github("GIScience/openrouteservice-r")
library(openrouteservice)
```

### #Cargo los datos de Excel

```
setwd("C:/Directorio de trabajo")
```

```
MatrizDistancias<-read_excel("MatrizDistancias.xlsx",col_names=TRUE)
```

```
Coordenadas<-read_excel("Coordenadas.xlsx",col_names = TRUE)
```

### #Convierto los datos cargados en matrices y establezco el rownames correspondiente

```
Distancias<-as.matrix(MatrizDistancias)
```

```
rownames(Distancias)<-colnames(Distancias)
```

```
Coordenadas<-as.matrix(Coordenadas)
```

```
coordenadas<-Coordenadas
```

```
ID1=rownames(coordenadas)<-rownames(Distancias)
```

```
coordenadas<-as.data.frame(coordenadas)
```

```
ui<-fluidPage(
```

```
  titlePanel("ALIMENTOS PARA TODOS"),
```

```
  sidebarLayout(
```

```
    sidebarPanel(
```

```
      selectInput("Origen",label="Almacen",
```

```
        choices=c("Berrioplano","Tudela"),selected=c("Berrioplano")),
```

```
      checkboxGroupInput("variable","Seleccionar las ciudades que se desea  
incluir en la ruta",
```

```
        choices=c("Berrioplano"="Berrioplano","Alsasua"="Alsasua","Barañain"="Barañain","Berriozar"="Berriozar","Erro"="Erro","Etxarri  
Aranarz"="Etxarri  
Aranarz","Irurtzun"="Irurtzun","Lekumberri"="Lekumberri","Noáin"="Noain","Pamplona"="Pamplona","Tafalla"="Tafalla","Uharte  
arakil"="Uharte  
arakil","Viana"="Viana","Tudela"="Tudela","Ablitas"="Ablitas","Arguedas"="Arguedas",  
,"Cadreita"="Cadreita","Caparroso"="Caparroso","Cascante"="Cascante","Cintruenigo"
```

```
= "Cintruenigo", "Cortes" = "Cortes", "Falces" = "Falces", "Fitero" = "Fitero", "Fustiñana" = "Fustiñana", "Marcilla" = "Marcilla", "Murchante" = "Murchante", "Murilo el cuende" = "Murilo el cuende",
```

```

"San Adrian" = "San Adrian", "Valtierra" = "Valtierra", "Villafranca" = "Villafranca"), selected = c("Berrioplano" = "Berrioplano", "Alsasua" = "Alsasua", "Barañain" = "Barañain", "Berriozar" = "Berriozar", "Noain" = "Noain", "Pamplona" = "Pamplona", "Tafalla" = "Tafalla", "Tudela" = "Tudela", "Cintruenigo" = "Cintruenigo",

```

```
"San Adrian" = "San Adrian"), inline = TRUE),
```

```
width = 12)
```

```

mainPanel(textOutput("Distancia"),
  verbatimTextOutput("values"),
  tableOutput("values"),
  leafletOutput("mapa")
))

```

```
)
```

```

server <- function(input, output) {
  slidervalues <- reactive({
    if(input$Origen %in% input$variable == TRUE) {
      MatrizTrabajo <- Distancias[c(input$variable), c(input$variable)]
    } else { MatrizTrabajo <- Distancias[c(input$Origen, input$variable),
      c(input$Origen, input$variable)] }
  })

```

```

if(length(input$variable) < 2)
  stop("Elegir mínimo dos ciudades para obtener ruta")

```



## Código completo empleado

```
else {
  MatrizTSP<-TSP(as.matrix(MatrizTrabajo), labels = NULL)
  initialtour<-as.integer(which(labels(MatrizTrabajo)[[1]]==input$Origen[1]))
  solucion1<-
solve_TSP((MatrizTSP),method="farthest_insertion",control=list(start=initialtour))
  solucion2<-
solve_TSP((MatrizTSP),method="nearest_insertion",control=list(start=initialtour))
  Longitud1<-tour_length(solucion1)
  Longitud2<-tour_length(solucion2)
  if(Longitud1<Longitud2){
    solucionRuta=solucion1
  }else{
    solucionRuta=solucion2
  }
  labels(solucionRuta)
}
})
slidervalues1<-reactive({
  if(input$Origen%in%input$variable==TRUE)
    MatrizTrabajo<-Distancias[c(input$variable),c(input$variable)]
  else MatrizTrabajo<-Distancias[c(input$Origen,input$variable),
    c(input$Origen,input$variable)]

  if(length(input$variable)<2)
    stop("Elegir minimo dos ciudades para obtener ruta")
  else {
    MatrizTSP<-TSP(as.matrix(MatrizTrabajo), labels = NULL)
    initialtour<-as.integer(which(labels(MatrizTrabajo)[[1]]==input$Origen[1]))
```

```
solucion1 <-
solve_TSP((MatrizTSP),method="farthest_insertion",control=list(start=initialtour))

solucion2 <-
solve_TSP((MatrizTSP),method="nearest_insertion",control=list(start=initialtour))

Longitud1 <-tour_length(solucion1)
Longitud2 <-tour_length(solucion2)
if(Longitud1 <Longitud2){
  solucionRuta=solucion1
} else{
  solucionRuta=solucion2
}
V <-as.integer(solucionRuta)
tour_length(MatrizTSP, V)
})

slidervalues2 <-reactive({
  if(input$Origen%in%input$variable==TRUE){
    MatrizTrabajo <-Distancias[c(input$variable),c(input$variable)]}
  else{ MatrizTrabajo <-Distancias[c(input$Origen,input$variable),
    c(input$Origen,input$variable)]}

  if(length(input$variable)<2)
    stop("Elegir minimo dos ciudades para obtener ruta")
  else {
    MatrizTSP <-TSP(as.matrix(MatrizTrabajo), labels = NULL)
    initialtour <-as.integer(which(labels(MatrizTrabajo)[[1]]==input$Origen[1]))
    solucion1 <-
solve_TSP((MatrizTSP),method="farthest_insertion",control=list(start=initialtour))
```

## Código completo empleado

```

solucion2<-
solve_TSP((MatrizTSP),method="nearest_insertion",control=list(start=initialtour))

Longitud1<-tour_length(solucion1)
Longitud2<-tour_length(solucion2)
if(Longitud1<Longitud2){
  solucionRuta=solucion1
}else{
  solucionRuta=solucion2
}

coordenadasok<-coordenadas[attributes(solucionRuta)$names,]
CoordenadasFase<-cbind(coordenasok[,1],coordenasok[,2])
CoordenadasFase<-rbind(CoordenadasFase,CoordenadasFase[1,])
coordenadasFasef<-function(x){ CoordenadasFase[x,]}
coordenadasFase<-
lapply(seq(1,dim(CoordenadasFase)[1]),coordenadasFasef)

ors_api_key('5b3ce3597851110001cf62485fce866458364a43b3aa39e829eef59c')

trayecto<-
ors_directions(coordenasFase,format="geojson",radiuses=rep(10000,length(coorde
nadasFase)))

ID1=rownames(coordenasok) <-rownames(coordenasok)

leaflet(data = coordenasok) %>%
  addTiles() %>%
  addMarkers(~LONGITUD, ~LATITUD, popup = ~as.character(ID1), label =
~as.character(ID1))%>%
  addGeoJSON(trayecto, fill=FALSE) %>%
  fitBBox(trayecto$bbox)
}

```

```
  })

  output$values<-renderTable({
    Tabla<-data.frame(slidervalues())
    names(Tabla)<-"Ruta de reparto"
    Tabla
  })

  output$Distancia<- renderText({
    "Longitud de la ruta(Km)"
  })

  output$valuess<-renderPrint({
    slidervalues1()
  })

  output$mapa<-renderLeaflet({
    slidervalues2()
  })
}

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

## ANEXO 2. MANUAL DE INSTALACIÓN DE LA APP

La instalación de la aplicación desarrollada en el presente trabajo fin de grado llamada "Alimentos para todos" requiere la instalación del software RStudio. Es un software que se puede descargar en la siguiente página web: [www.rstudio.com/](http://www.rstudio.com/).

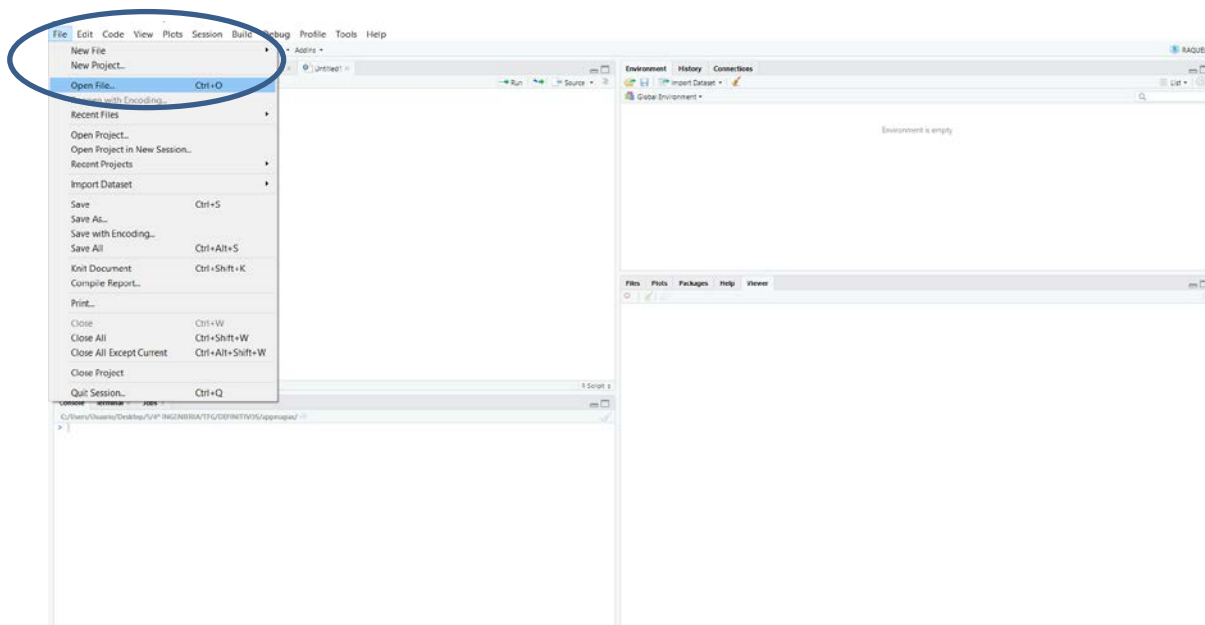
Se deben instalar los paquetes: Shiny, TPS, Leaflet, readXL y OpenRouteService.

Se debe estar en posesión de los archivos necesarios para la generación de la aplicación. Dichos archivos serán proporcionados mediante el correo 743405@unizar.es al que deben dirigirse para la obtención de los mismos explicando el futuro uso de la aplicación.

Los archivos necesarios son los siguientes: uiserver.R, Aplicación.RData, RA.png.

Se creará una carpeta que de ahora en adelante llamaremos TFG dónde estarán los archivos de uiserver.R y Aplicación.RData. Dentro de esta carpeta habrá otra que contenga la imagen RA.png, la cual nombraremos a partir de ahora como IMG.

Una vez la carpeta descrita y llamada TFG se encuentre en el destino deseado del CPU, se iniciará el software RStudio y se cargará el archivo uiserver.R. Para cargar dicho archivo desde RStudio, se debe ir a File y clicar sobre Open File, para buscar el archivo y seleccionarlo.



**Ilustración 1. Cargar archivos desde RStudio.**

A continuación se debe cargar el espacio de trabajo. Para ello se escribe en la consola la función `setwd()` y se indica entre los paréntesis la ruta hasta la carpeta TFG.

Finalmente sólo queda darle al botón de Run App y se abrirá una ventana con la interfaz de la aplicación lista para usarla.

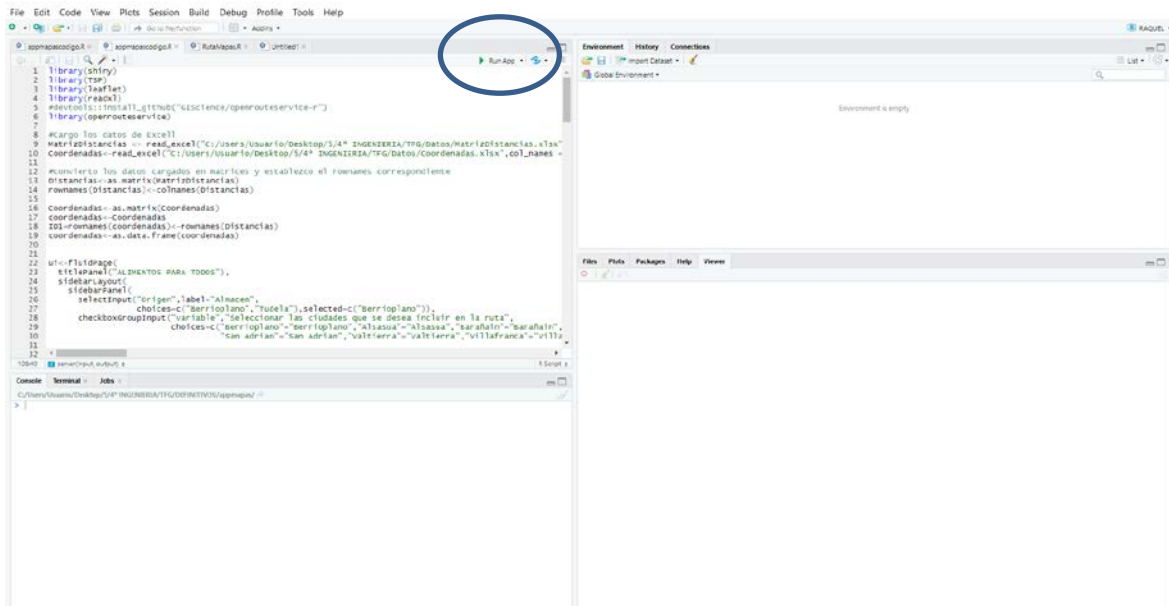


Ilustración 2. Correr aplicación.

Matriz de distancias

## ANEXO 3. MATRIZ DE DISTANCIAS

Tabla 1. Matriz de distancias.

Berrioplano	Alsasua	Barañain	Berriozar	Erro	Etxarri Aranz	Irurtzun	Lekumberri	Noain	Pamplona	Tafalla	Uhartearakil	Viana	Tudela	Ablitas	Arguedas	Cadreita	Caparroso	Cascante	Cintruenigo	Cortes	Falces	Fitero	Fustiñana	Marcilla	Murchante	Murilo el cuende	San Adrian	Valtierra	Villafranca
0	45,2	8,8	3,1	34	36,7	15,2	29,6	18	7,2	49,5	30	89	110	114	94,7	86,4	69,2	115	105	129	70	110	119	72,5	108	63,3	85,9	87,2	77,8
45,2	0	53,3	47,9	78	10,9	31,2	43,4	61	51,7	93	19,3	100	152	156	128	117	108	150	135	158	91	127	159	104	147	102	88,8	121	109
8,8	53,3	0	6,3	38	43,1	23,1	36	9,4	5,1	42,2	36,4	82	101	105	85,8	77,4	60,2	107	96	120	59	101	110	63,6	99	54,4	69,1	78,3	68,9
3,1	47,9	6,3	0	30	39,7	16,6	32,6	17	4,6	49	33	88	109	113	94,1	85,7	68,5	115	104	128	69	110	119	71,8	107	62,6	85,3	86,6	77,2
33,8	78,3	38	30,4	0	68,9	48,9	61,8	40	29,3	70,6	62,1	115	133	130	115	108	84,3	132	121	152	85	126	142	87,6	124	78,4	102	107	93,9
36,7	10,9	43,1	39,7	69	0	21,1	33,4	51	41,6	83	9,2	89	143	144	130	108	101	141	127	149	82	119	150	95	139	96,9	80	122	99,8
15,2	31,2	23,1	16,6	49	21,1	0	14,2	31	21,1	62,5	16	102	123	128	108	100	82,8	129	119	143	82	124	133	86,1	122	76,9	99,6	101	91,5
29,6	43,4	36	32,6	62	33,4	14,2	0	45	35,1	76,4	28	116	136	141	121	113	95,7	142	131	155	97	137	146	99	134	89,8	112	114	104
17,9	61,4	9,4	17,4	40	51,4	30,9	44,8	0	6,3	34,5	45,1	86	92,8	97	77,7	69,3	52,1	98,4	87,9	112	53	93	97,8	55	90,9	46,2	71,4	70,2	60,8
7,2	51,7	5,1	4,6	29	41,6	21,1	35,1	6,3	0	38,4	35,2	82	98,4	103	83,3	75	57,8	104	93,5	118	59	99	108	60,6	96,5	51,9	70,5	75,9	66,4
49,5	93	42,2	49	71	83	62,5	76,4	35	38,4	0	74,1	75	63,6	67,8	48,1	39,7	22,5	69	58,7	82,7	24	64	68,2	26,3	61,7	16,7	41,8	40,6	31,6
30	19,3	36,4	33	62	9,2	16	28	45	35,2	74,1	0	114	137	141	122	113	96	142	132	156	93	137	146	99,4	135	90,2	85,6	114	105
88,7	99,6	81,5	88,2	115	89,3	102	116	86	82,2	75,4	114	0	98,8	102	88	75,4	74,7	94,9	82,5	118	70	86	108	65,2	95,1	79,7	48,7	80,5	79,9
110	152	101	109	133	143	123	136	93	98,4	63,6	137	99	0	12,3	14,8	23,6	37,4	12,1	19,2	24,6	48	25	14,3	39	7,2	44,6	48,4	19,1	31,9
114	156	105	113	130	144	128	141	97	103	67,8	141	102	12,3	0	28,9	33,3	47,2	4,6	22,6	23,6	56	24	16,9	46,6	9,4	54,2	56,1	33,2	41,6
94,7	128	85,8	94,1	115	130	108	121	78	83,3	48,1	122	88	14,8	28,9	0	13	26,7	28,6	29,6	37,2	39	36	24	28,4	26,4	33,9	41,2	4,8	21,3
86,4	117	77,4	85,7	108	108	100	113	69	75	39,7	113	75	23,6	33,3	13	0	18,4	34,1	22,6	46,1	26	28	32,9	15,8	26,4	25,5	28,6	5,5	8,7
69,2	108	60,2	68,5	84	101	82,8	95,7	52	57,8	22,5	96	75	37,4	47,2	26,7	18,4	0	47,5	36,1	59,6	19	42	46,3	9,6	39,9	8,3	31,1	18,9	14,6
115	150	107	115	132	141	129	142	98	104	69	142	95	12,1	4,6	28,6	34,1	47,5	0	18,1	28	57	20	21,2	47,8	5,7	55,7	49,7	32,6	40,8
105	135	96	104	121	127	119	131	88	93,5	58,7	132	83	19,2	22,6	29,6	22,6	36,1	18,1	0	40,8	47	5,8	31,4	37,7	17	43,4	37,3	22,8	30,8
129	158	120	128	152	149	143	155	112	118	82,7	156	118	24,6	23,6	37,2	46,1	59,6	28	40,8	0	71	47	16,2	65,4	29,1	69,2	71	42,1	54,9
70,1	91,2	58,9	69,4	85	82,4	82,2	96,6	53	58,7	23,5	93,1	70	48,4	56,2	38,6	26	18,5	57,2	47,2	71	0	48	59,5	12,4	49,4	23,5	26,7	32,1	17,5
110	127	101	110	126	119	124	137	93	98,99	64,1	137	86	24,6	24,2	35,8	28	41,5	19,8	5,8	46,5	48	0	36,8	43,2	22,5	48,8	40,7	28,2	36,2
119	159	110	119	142	150	133	146	98	108	68,2	146	108	14,3	16,9	24	32,9	46,3	21,2	31,4	16,2	60	37	0	51,9	23,1	54	61,3	28,6	41,4
72,5	104	63,6	71,8	88	95	86,1	99	55	60,6	26,3	99,4	65	39	46,6	28,4	15,8	9,6	47,8	37,7	65,4	12	43	51,9	0	40,2	14,8	21,7	22,5	8,8
108	147	99	107	124	139	122	134	91	96,5	61,7	135	95	7,2	9,4	26,4	26,4	39,9	5,7	17	29,1	49	23	23,1	40,2	0	48,2	50	27,6	35,6
63,3	102	54,4	62,6	78	96,9	76,9	89,8	46	51,9	16,7	90,2	80	44,6	54,2	33,9	25,5	8,3	55,7	43,4	69,2	24	49	54	14,8	48,2	0	36,1	26,4	19,7
85,9	88,8	69,1	85,3	102	80	99,6	112	71	70,5	41,8	85,6	49	48,4	56,1	41,2	28,6	31,1	49,7	37,3	71	27	41	61,3	21,7	50	36,1	0	32,9	32,2
87,2	121	78,3	86,6	107	122	101	114	70	75,9	40,6	114	81	19,1	33,2	4,8	5,5	18,9	32,6	22,8	42,1	32	28	28,6	22,5	27,6	26,4	32,9	0	13,8
77,8	109	68,9	77,2	94	99,8	91,5	104	61	66,4	31,6	105	80	31,9	41,6	21,3	8,7	14,6	40,8	30,8	54,9	18	36	41,4	8,8	35,6	19,7	32,2	13,8	0

## ANEXO 4. TABLA DE COORDENADAS DE MUNICIPIOS

	LONGITUD	LATITUD
Berrioplano	-1,7	42,86
Alsasua	-2,16	42,89
Barañain	-1,68	42,8
Berriozar	-1,67	42,83
Erro	-1,44	42,94
Etxarri Aranarz	-2,06	42,9
Irurtzun	-1,82	42,91
Lekumberri	-1,89	43
Noain	-1,63	42,76
Pamplona	-1,64	42,81
Tafalla	-1,67	42,52
Uhartelarakil	-1,96	42,91
Viana	-2,37	42,51
Tudela	-1,6	42,06
Ablitas	-1,63	41,97
Arguedas	-1,59	42,17
Cadreita	-1,69	42,21
Caparroso	-1,65	42,33
Cascante	-1,67	41,99
Cintruenigo	-1,8	42,07
Cortes	-1,41	41,92
Falces	-1,79	42,38
Fitero	-1,85	42,05
Fustiñana	-1,48	42,02
Marcilla	-1,73	42,32
Murchante	-1,65	42,03
Murilo el cuende	-1,63	42,39
San Adrian	-1,93	42,33
Valtierra	-1,63	42,19
Villafranca	-1,74	42,27

Tabla 2. Listado de coordenadas.





## Relación de documentos

<input type="checkbox"/> Memoria .....	NN	páginas
<input checked="" type="checkbox"/> Anexos .....	17	páginas

La Almunia, a 22 de Junio de 2020

Firmado: Raquel Escalera Lapuerta