



Universidad Pública de Navarra  
Nafarroako Unibertsitate Publikoa

Departamento de Estadística, Informática y Matemáticas

Trabajo de fin de grado

# **Plataforma de adquisición de datos pluviométricos para la predicción y aviso de inundaciones**

**Autor:**  
**Arkaitz Oderiz Garin**

Supervisor:  
Unai Pérez-Goya

Pamplona, 2022



## Contenidos

<b>Lista de Figuras</b>	<b>iii</b>
<b>Lista de Tablas</b>	<b>iv</b>
<b>Lista de Códigos</b>	<b>vi</b>
<b>Lista de Teoremas</b>	<b>viii</b>
<b>Lista de Definiciones</b>	<b>x</b>
<b>Lista de Lemas</b>	<b>xii</b>
<b>1 Introducción</b>	<b>1</b>
1.1 Esto es una sección . . . . .	1
1.1.1 Esto es una subsección . . . . .	1
<b>2 Tecnologías</b>	<b>3</b>
2.1 Debian . . . . .	3
2.1.1 Historia . . . . .	3
2.1.2 Filosofía . . . . .	3
2.1.3 Modelo de negocio . . . . .	4
<b>3 Algoritmo</b>	<b>5</b>
<b>4 Evaluación</b>	<b>7</b>
<b>5 Conclusiones y Trabajo Futuro</b>	<b>9</b>
<b>Referencias</b>	<b>11</b>

<b>Glosario</b>	<b>13</b>
<b>Anexos</b>	<b>15</b>

## Lista de Figuras

1.1	Nombre reducido para tabla de figuras . . . . .	1
-----	---	---



## Lista de Tablas

1.1	Mi tabla de ejemplo . . . . .	2
-----	-------------------------------	---





## Lista de Códigos

3.1	pie de código . . . . .	5
A1	Mostrar notas. . . . .	15



**Lista de Teoremas**

1	Teorema (Sum) . . . . .	5
2	Teorema (About $C^1(0, 1)$ ) . . . . .	5



## Lista de Definiciones

1	Definición (Nice numbers) . . . . .	5
---	-------------------------------------	---



## Lista de Lemas

1	Lema . . . . .	5
---	----------------	---





## 1.1 Esto es una sección

Aquí tenemos una imagen referenciada 1.1. Una dirección [Mi dirección](#)<sup>1</sup>.

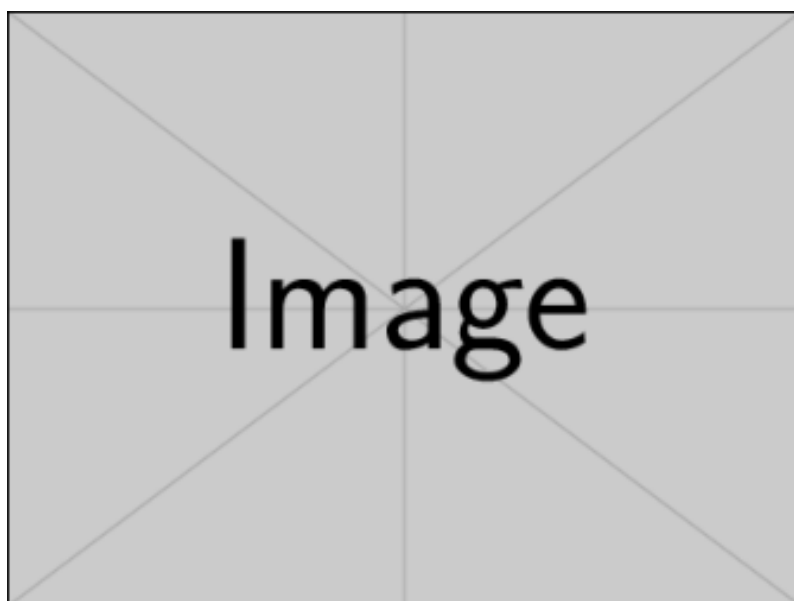


Figura 1.1: Real caption

Para generar entradas en el índice de palabras. SATA.

### 1.1.1 Esto es una subsección

Una lista de parámetros:

---

<sup>1</sup><https://www.copernicus.eu/es>

- uno.
- dos.
- tres.

Una lista enumerada

1. uno.
2. dos.

Vamos a citar ... [\[1\]](#)

Tabla 1.1: Mi tabla de ejemplo

Nombre	Medida	Otra cosa
10	10	4

## 2.1 Debian

### 2.1.1 Historia

Conforme la computación fue tomando terreno tanto en el ámbito comercial como en el escolar, no tener una forma de instalar y configurar los sistemas de una forma rápida sin tener que partir de cero y sin tener que compilar el software necesario manualmente se convirtió en un problema patente entre los usuarios.

En 1993, tras varios intentos fallidos por distintas empresas de solucionar el problema, Ian Murdock, por aquel entonces estudiante de la Universidad Purdue, encontró la solución al problema basándose en el reciente proyecto de Linus torvalds, el kernel Linux. tras el anuncio de Ian para crear un sistema operativo de forma descentralizada en paralelo como es el caso del kernel Linux, docenas de usuarios se unieron para formar el proyecto Debian Linux con la intención de crear un sistema operativo de gran calidad y mantenimiento, publicando en enero de 1994 la primera versión de Debian 0.91.

### 2.1.2 Filosofía

Debian no intenta seguir ni competir con los líderes del sector, por el contrario, desde sus inicios el proyecto se ha basado en una filosofía centrada en la robustez y estabilidad del sistema guiada por unos estrictos estándares de calidad, actualizándose conforme las necesidades de sus usuarios a la vez que promueve el software gratuito, lo que le ha ayudado a obtener fama entre los usuarios.

A su vez, debido al apoyo del software libre, hace uso de múltiples licencias de software como la Licencia Pública General GNU (GPL), licencias artísticas o del tipo BSD, lo cual ha llevado al desarrollo de las Directrices de Software Libre de Debian (DFSG) con el fin de definir la construcción del software libre.

Definido por estas licencias, el software libre debe cumplir al menos las que están consideradas la cuatro libertades esenciales:

- Ejecutar el programa como se desee, con cualquier propósito.
- Estudiar cómo funciona el programa, y cambiarlo para que haga lo que se desee.
- Redistribuir copias para ayudar a otros.
- Distribuir copias de sus versiones modificadas a terceros permitiendo ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones.

### **2.1.3 Modelo de negocio**

Código en R del ejemplo 3.1:

```

1  # Establecer las credenciales de la API [1]
2
3  library(rsat)
4  set_credentials("rsat.package", "UpnaSSG.2021")
5
6  #Definir la region
7
8  library(raster)
9  dir.create( "./RSATprueba/countries" ,recursive = TRUE)
10 spain<-getData('GADM', country= 'Spain', path=
    "./RSATprueba/countries", level=2)

```

Código 3.1: pie de código

**Teorema 1** (Sum).  $1 + 1 = 2$

**Definición 1** (Nice numbers). *A number is nice if it looks beautiful.*

**Teorema 2** (About  $C^1(0,1)$ ). *The set  $C^1(0,1)$  is interesting.*

Teorema en referentzia 2

*Proof.* To prove it by contradiction try and assume that the statement is false, proceed from there and at some point you will arrive to a contradiction.  $\square$

**Lema 1.** *To prove it by contradiction try and assume that the statement is false, proceed from there and at some point you will arrive to a contradiction.*

Lema en referentzia 1

$$1 + e^{i\pi} = 0. \quad (3.1)$$

Formula en referentzia 3.1









## Conclusiones y Trabajo Futuro



## Referencias

- [1] L. Lorenzi, F. Melgani, and G. Mercier, "Inpainting strategies for reconstruction of missing data in vhr images," *IEEE Geoscience and remote sensing letters*, vol. 8, no. 5, pp. 914–918, 2011.



## Glosario

parabras, 1

SATA, 1



```
1      public void mostrarTodasNotas(){
2          int a = 1;
3          System.out.println("Estas son todas las notas que hay
4                          guardadas.");
5          for(int j=0; j<lista_notas.size(); j++){
6              System.out.print(a+"-");
7              System.out.println(lista_notas.get(j).toString());
8              a++;
9          }
10     }
```

Código A1: Mostrar notas.