



Universidad Nacional Experimental del Táchira

Vicerectorado Académico

Decanato de Docencia

Departamento de Ingeniería en Informática

Trabajo de aplicación profesional

Proyecto especial de grado

Autor(es): Jesús David Escalante Rodríguez

C.I.: 21.220.841

jesusd.escalante@unet.edu.ve

Tutor(es): Rossana Timaure

rttg@unet.edu.ve

San Cristóbal, Julio de 2018.2018

Introducción

El siguiente documento se dividió en tres capítulos en los cuales se describe la propuesta de la investigación a desarrollar, los cuales estan estructurados de la siguiente forma:

Capítulo 1. Preliminares.

Capítulo 2. Marco teórico.

Capítulo 2. Metodología.



Universidad Nacional Experimental del Táchira
Vicerrectorado Académico
Decanato de Docencia
Departamento de Ingeniería Informática
Trabajo de Aplicación Profesional
Proyecto Especial de Grado

**Aprobación del Tutor para presentación de la Propuesta del Proyecto Especial de
Grado**

Yo, Rossana Timaure titular de la cédula de identidad No. V-12.021.405. en mi carácter de Tutor(a) del Proyecto Especial de Grado titulado: , presentado por el bachiller: titular de la cédula de identidad No. , por medio de la presente autorizo la presentación de la Propuesta de Proyecto Especial de Grado ante los jurados designados por la comisión de Trabajo de Aplicación Profesional del Departamento de Ingeniería en Informática, en virtud de considerar que reúne los requisitos establecidos en el artículo 16 de las Normas para el Trabajo de Aplicación Profesional de la UNET.

Nombres y apellidos del tutor
C.I. V- 12.021.405
Rossana Timaure

Capítulo 1

Preliminares

1.1. Planteamiento y formulación del problema

1.2. Objetivos

1.2.1. Objetivo General

1.2.2. Objetivos Específicos

-

1.3. Justificación e Importancia

1.4. Alcance y Limitaciones

Capítulo 2

Fundamentos teóricos

2.1. Antecedentes

2.2. Bases Teóricas

2.3.

1.

L

2.3.1. Lenguaje R.

R es un conjunto integrado de *software* de código abierto para el almacenamiento, manipulación, cálculo y visualización de datos para computación y gráfica estadística, puede ser compilado y ejecutado en Windows, Mac OS X y otras plataformas UNIX (como Linux), se distribuye usualmente en formato binario (<https://www.r-project.org/about.html>, 2018). El proyecto de *software* R fue iniciado por Robert Gentleman y Ross Ihaka. El lenguaje fue influenciado por lenguaje S desarrollado originalmente en Bell Laboratories por John Chambers y sus colegas. Desde entonces ha evolucionado para el cálculo estadístico asociado a diversas disciplinas para contextos académicos y comerciales. En R, la unidad fundamental de código compartible es el paquete, el cual agrupa código, datos, documentación y pruebas, y resulta simple de compartir con otros. Para enero del 2015 ya habían más de 6.000 paquetes disponibles en la Red Integral de Archivos de R, conocido comunmente por su acrónimo CRAN, el cual es el repositorio de paquetes . Esta gran variedad de paquetes es una de las razones por las cuales R es tan exitoso, pues es probable que

algún investigador o académico ya haya resuelto un problema en su propio campo usando esta herramienta, por lo que otros usuarios simplemente podrán recurrir a ella para su uso directo o para llamarla en un nuevo código (Wickham,2015).

2.3.2. RStudio.

RStudio es un ambiente de desarrollo integrado (*Integrated Development Environment*, IDE) que ofrece herramientas de desarrollo vía consola, editor de sintaxis que apoya la ejecución de código, así como herramientas para el trazado, la depuración y la gestión del espacio de trabajo. RStudio está disponible para Windows, Mac y Linux o para navegadores conectados a RStudio Server o RStudio Server Pro (Debian / Ubuntu, RedHat / CentOS, y SUSE Linux) (<https://www.rstudio.com/about/>, 2018)

2.3.3. Estructura de paquetes en R/RStudio.

Requerimiento del núcleo (*core*)

1. DESCRIPTION: metadatos del package .

La tarea del archivo Description es de gran importancia ya que es en el donde se registra la metadata, las dependencias que utiliza el paquete, la licencia y el soporte en caso de ocurrir errores con el mismo La estructura mínima para realizar un paquete en R es la siguiente:

- Package: mypackage
- Title: What The Package Does (one line, title case required)
- Version: 0.1
- Authors@R: person("First", "Last", email = "first.last@example.com",
role = c("aut", "cre"))
- Description: What the package does (one paragraph)
- Depends: R (*i*= 3.1.0)
- License: What license is it under?
- LazyData: true

2. **R/**: dirección del repositorio donde se encuentra el código del paquete (.R files).

Se expondrán las buenas prácticas a la hora de realizar todo nuestro código en R, desde organización de las funciones, estilos de código y nombre de variables

Organizar funciones en R: aunque puedes organizar los archivos como desees, los dos extremos son malos no colocar todas las funciones en el mismo. archivo y no crear un archivo para para función, aunque si una función es muy grande o tiene mucha documentación se puede dar el caso, los nombres de los archivos tienen que ser significativo y deben de terminar en R

- Bien
 - fit_models.R
 - utility_functions.R
- Mal
 - foo.r
 - stuff.r

Se puede recomendar de acuerdo al número de función utilizar prefijo

Nombres de Objetos: Los nombres de las Variables y funciones deben de ser en minúsculas, usar el guión bajo (-) para separar palabras

- Bien
 - day_one
 - day_1
- Mal
 - first_day_of_the_month
 - DayOne
 - dayone
 - djm1

En lo posible no usar nombres de variables existentes esto causará confusión.

Espaciado: Se recomienda colocar espacios alrededor de todos los operadores lógicos y aritméticos (=, +, -, <-, etc.). siempre coloque un espacio después de una coma, y nunca antes de ella.

- Bien

- `average <- mean(feet / 12 + inches, na.rm = TRUE)`

- Mal

- `average<-mean(feet/12+inches,na.rm=TRUE)`

Hay una pequeña excepción a esta regla: (`:`, `::` y `:::`) no necesitan espacios alrededor de ellos.

- Bien

- `x <- 1:10`
- `base::get`

- Mal

- `x <- 1 : 10`
- `base :: get`

Dejar un espacio antes del paréntesis izquierdo, excepto en la llamada a una función

- Bien

- `if (debug) do(x)`
- `plot(x,y)`

- Mal

- `if(debug)do(x)`
- `plot(x, y)`

Se Utiliza mas de un espacio es caso que de mejore a la alineación, por ejemplo:

```
list(  
  total  = a + b + c,  
  mean   = (a + b + c) / n  
)
```

No coloque espacios alrededor del código entre paréntesis o corchetes (a menos que haya una coma)

■ Bien

- `if (debug) do(x)`
- `diamonds[5,]`

■ Mal

- `if (debug) do(x)` *# no espacios alrededor de debug*
- `x[1,]` *# necesita un espacio despues de la coma*
- `x[1 ,]` *# el espacio va despues de la coma no antes*

Llaves: Una llave de apertura nunca debe ir en su propia línea y siempre debe ir seguida de un nueva línea. Una llave siempre debe ir en su propia línea, a menos que sea seguida por otra y siempre sangría el código dentro de las llaves

■ Bien

- ```
if (y < 0 && debug) {
 message("Y es negativo")
}
```
- ```
if (y == 0) {  
    log(x)  
} else {  
    y ^ x  
}
```

■ Mal

- ```
if (y < 0 && debug)
message("Y es negativo")
```
- ```
if (y == 0) {  
    log(x)  
}  
else {  
    y ^ x  
}
```

Sentencias muy cortas bien dejarlas en la misma línea.

```
if(y < 0 && debug) message("Y es negativo")
```

Longitud de Línea: cada línea debe de llevar máximo 80 caracteres, si se queda sin espacio es recomendable utilizar una función separada

Sangría: Utilice sangría de 2 espacios, nunca use tabulador o múltiples tabuladores o espacios. La única excepción es cuando se define una sentencia en múltiples líneas.

```
long_function_name <- function(  a = "a long argument",  
                                  b = "another argument",  
                                  b = "another argument",
```

Asignación: Usar el `<-`, y no `=`

- Bien

- `x <- 5`

- Mal

- `x = 5`

3. `man/`: documentación.

4. `NAMESPACE`: especifica que objetos conforman el paquete.

Comentarios: Comente tu código, el comentario comienza `#`, los comentarios deben de explicar el porque, no el que.

use los caracteres `(-)` y `(=)` para separar líneas

```
# Load data - - - - -
```

```
# Plot data - - - - -
```

2.3.4. Glosario

R :

Capítulo 3

Fundamentos Metodológicos

A continuación se plantea la estructura a seguir por el presente trabajo, detallando el enfoque, tipo, nivel y diseño de la investigación y la metodología a implementar entre otros.

3.1. Enfoque de la investigación

La presente investigación se desarrollará siguiendo un enfoque cuantitativo, puesto que, como lo indican Pallela y Martins (2012) , “la investigación cuantitativa requiere el uso de instrumentos de medición y comparación, que proporcionan datos cuyo estudio necesita la aplicación de modelos matemáticos y estadísticos, el conocimiento está basado en hechos”. Los datos a usar en el desarrollo del paquete provienen de ensayo experimental desarrollado en los Laboratorios de Biofertilizante e Instrumentación Biológica, adscritos al decanato de Investigación de la Universidad Nacional Experimental del Táchira .

3.2. Tipo o nivel de investigación

Este proyecto plantea un tipo de investigación de campo, según como lo indican Pallela y Martins (2012), la investigación de campo “consiste en la recolección directamente de la realidad donde ocurren los hechos, sin manipular o controlar variables” ya que permite indagar los efectos de la interrelación entre los diferentes tipos de variable en lugar de los hechos.

En este punto se debe determinar la profundidad que abarca esta investigación, teniendo en cuenta que de acuerdo con el nivel de la investigación es definido como “grado de profundidad con

que se aborda un fenómeno u objeto de estudio” (Arias, 2012).

En este sentido, se tiene que dadas las características del proyecto, se asocia con un nivel descriptivo, tal como lo indican Pallela y Martins (2012), “hace énfasis sobre conclusiones dominantes o sobre como una persona, grupo o cosa se conduce o funciona en el presente” esto debido a que se medirán los datos extraídos sin alterarlos para ser mostrados en el sistema.

Cuando se habla de un nivel descriptivo junto con una investigación de tipo de campo, en ella no se formulan hipótesis y las variables se enuncian en los objetivos de la investigación que se desarrollará.

3.3. Diseño de la investigación

Según Arias(2012), el diseño de la investigación es “la estrategia general que adopta el investigador para responder al problema planteado” (p.21) por lo que es vital establecer una correcta secuencia de pasos para elaborar el prototipo de software que dará solución a la problemática principal de la investigación.

Con este enfoque, se tiene que este trabajo seguirá un diseño no experimental, enfocado en el uso de información existente, de acuerdo con lo dicho por Pallela y Martins (2012) al definir el diseño no experimental como:

Es el que se realiza sin manipular en forma deliberada ninguna variable. El investigador no sustituye intencionalmente las variables independientes. Se observan los hechos tal y como se presentan en su contexto real y en un tiempo determinado o no, para luego analizarlos. Por lo tanto, este diseño no se construye una situación específica sino que se observan las que existen. Las variables independientes ya han ocurrido y no pueden ser manipuladas, lo que impide influir sobre ellas para modificarlas. (p.81)

Esto indica que no hay manipulación de variables. Esta investigación presenta una modalidad de proyecto especial que, como lo indican Pallela y Martins (2012), los proyectos especiales “destinados a la creación de productos que puedan solucionar deficiencias evidenciadas, se caracterizan por su valor innovador y aporte significativo” (p.92), ya que se creará un *software* aplicable al área de estudio.

3.4. Metodología

Para el desarrollo del paquete se seguirán las pautas estándar establecidas para la creación de paquetes y extensiones en R.

Creación del esqueleto del paquete.

En esta etapa se diseñaran y crearan los directorios, ficheros y objetos que conformaran el paquete.

Registrar el método para el envío y uso de funciones.

Es esta etapa del desarrollo se estableceran las dependencias sobre los paquetes de la base fuente de código R y sus métodos de conexión, considerando el manejo de versiones y los criterios de mantenimiento, además se estableceran los espacios de nombre o las estrategias para la búsqueda y utilización de las variables; unificando estos criterios a las funciones que seran diseñadas.

Diseño y codificación de las funciones.

Los métodos para el diseño de las funciones primarias en R serán los diagramas de flujo; y para su codificación se seguirán las normas de estilo para codificación en R, sugeridas por Wickham (2015) y por el creador del paquete *formatR* Xie(2017), además se establecerán la dependencia con las funciones de código base y las recomendadas para desarrollo en R.

Pruebas unitarias de las funciones.

Debido a que los paquetes en R están conformados, entre otros elementos por las funciones primarias, a cada una de ellas se les realizaran pruebas unitarias en dos fases, la primera con datos sintéticos que permitan comprobar cada estado del diagrama de flujo, que esquematiza la solución numérica que permite el cálculo de cada uno de los índices fisiológicos para entre otra herramientas pueden utilizarse los paquetes de *RUnit* (Zenka, 2015) y *testthat* (Wickham, 2017), y la segunda etapa donde cada función asociada a un índice fisiológico se le realizan prueban

con los conjuntos de datos de prueba que formaran parte integral del paquete y con los cuales se desarrollarían los ejemplos prácticos que conformarían la documentación que acompañaría al paquete R.

Chequear la carga del paquete.

En esta etapa del desarrollo se utilizaran las funciones de chequear paquete que ofrece el código R; cuya finalidad es verificar cada fichero del árbol de carpetas asociadas a cada elemento de la estructura o esqueleto del paquete, que a su vez creara el archivo de documentación en LaTeX y/o HTML, compilará el código fuente y creará las librerías de enlace dinámico (*dynamic link library* DLL).

Construcción del método de distribución del paquete.

Se seleccionara la forma de distribución del paquete desde el repositorio local, creando los ficheros fuentes (en formato *tarball*) y en binario.

3.5. Aspectos administrativos

La realización de la investigación será planificada según lo establecido en el siguiente diagrama:

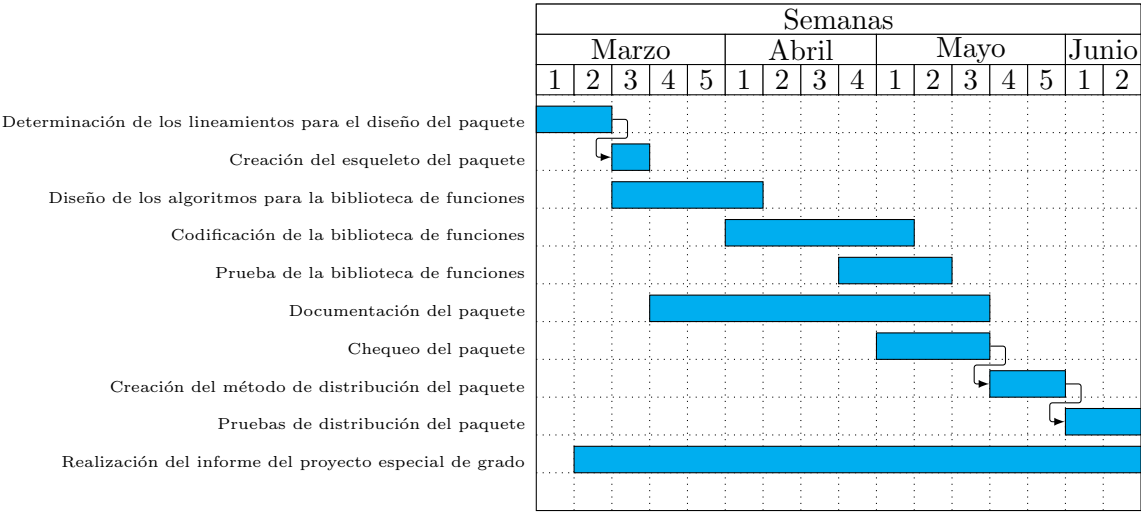


Figura 3.1: Diagrama de Gantt con la planificación del proyecto especial de grado

Referencias Bibliográficas

Available CRAN Packages By Name. Disponible en: https://cran.r-project.org/web/packages/available_packages_by_name.html. Consultada Enero 2018.

Middle Sinu Valley, Colombia. Agronomía Colombiana 22 (1): 81-90

Chambers, John M (1998). Programming with Data: A Guide to the S Language. Springer.

Filippaa,G., Cremonesea,E., Migliavacca, m., Galvagnoa, M., Forkel,M., Wingate,L., Tomelleri,E., Morra,U., Richardsone,A.D. (2016). Phenopix: A R package for image-based vegetation phenology. Agricultural and Forest Meteorology 220, 141-150.

Leisch, F. (2002), “Sweave, Part I: Mixing R and LATEX,” R News, 2, 28–31, URL <http://CRAN.R-project.org/doc/Rnews/>.

Leisch F. (2009). Creating R Packages: A Tutorial.

Marsden J., Weinstein A. (1985). Calculus I. Springer-Verlag, New York Inc.

morphological index. Forest Science and Technology, Vol. 11, No. 1, 1-10.

Trancón B., Carl W., Bolz F., Grelck C. (2012). The Functional Programming Language R and the Paradigm of Dynamic Scientific Programming. (182-197).

Wickham H. (2015). R Packages: Organize, Test, Document, and Share Your Code. 1 Edic. Hadley – Wickham.

Xie Y. (2017) . Package 'formatR'. Disponible en:<https://cran.r-project.org/web/packages/formatR/formatR.pdf>. Consultado Enero 2018.

Ablan, M. Márquez, R. Rivas, Y. Molina, A y Querales, J. (2011, Noviembre). Una librería en R para validación de modelos de simulación. Revista Ciencia e Ingeniería.” , Volumen:(Edición Especial: “Jornada de Modelado y Simulación”), pp.117-126.

Yeater, K. Duke, S and Riedell, W. (2015, Febrero 25). Multivariate Analysis: Greater Insights into Complex Systems. Agronomy Journal , Volumen(107), pp.799-810

Fernández Contreras, M.(2017). Modelo de Forrester para evaluar procesos biológicos, casos de estudio: crecimiento de hongos en medios líquidos y crecimiento de plántulas de pimentón *Capsicum annum* L. biofertilizadas. (Trabajo Especial de Grado de pregrado).Universidad Nacional Experimental del Táchira. San Cristóba, Estado Táchira.