Universidad Nacional Experimental del Táchira

Vicerectorado Académico

Decanato de Docencia

Departamento de Ingeniería en Informática

Trabajo de aplicación profesional

Proyecto especial de grado

# Paquete en lenguaje R, para la simulacion del modelo MOMOS

Autor: Ghabriel Villareal

C.I.:22.676.954

ghabriel.villarreal@unet.edu.ve

Tutor(es): Rossana Timaure

rttg@unet.edu.ve

San Cristóbal, Octubre de 2022

# Capítulo 1

# El Problema

## 1.1. Planteamiento y formulación del problema

La materia orgánica estable del suelo (MOS) se caracteriza por ser el resultado del proceso de descomposición de los residuos de animales y vegetales, los cuales varían en proporción y estado. Estos residuos son sustancias que contribuyen a la fertilidad del suelo, por lo que, la materia orgánica del suelo MOS, viene a ser uno de los factores de importancia que determina localidad de un suelo.

La descomposición de la materia orgánica es un proceso biológico que ocurre naturalmente, por efecto de diferentes grupos de organismos, entre los cuales destacan los microorganismos del suelo, que permiten transformar los diferentes residuos, en nutrientes básicos y energía. Siendo que una de las funciones MOS es actuar como depósito de carbono, permitiendo que exista en el suelo más del doble del carbono contenido en la atmósfera, por lo que los cambios que experimente está, pueden tener un impacto en el equilibrio global, y por lo tanto un efecto sobre las condiciones climáticas del planeta, al ser considerado el carbono que sale de esta como CO2, el cual es uno de los gases que contribuyen al efecto invernadero del planeta y al cambio climático (IPCC, 2000).

En la actualidad, se reporta un incremento en el cambio climático por efecto del impacto antropogénico, el cual se observa en el aumento de la temperatura en diferentes regiones, y como efecto de retroalimentación la temperatura influye sobre el proceso de descomposición y por lo tanto en el almacenamiento de carbono en el suelo y en la vegetación (Andriulo, 2006), todo esto muestra que el suelo es un compartimiento complejo debido a todas las interrelaciones que existen en el mismo.

Por lo que la comprensión de los diferentes procesos y transformaciones que ocurren en el suelo, son de gran importancia, sin embargo la cantidad de información que puede ser recolectada, así como la variabilidad en la misma, constituyen un desafío en sí, por lo cual esto conlleva ha utilizar herramientas tecnológicas que permitan el manejo de la información obtenida, es por ello que existe una gran cantidad de modelos que simulan y describen los procesos de la descomposición de la MOS, y los cuales pueden o están siendo acoplados a modelos de pronósticos a nivel global, de igual manera estos modelos están evolucionando, en la búsqueda de una mejor comprensión de los diversos procesos físico, químicos y biológicos [Larionova et al., 2007; Lomander et al., 1998].

Existen una gran cantidad de modelos que estudian la MOS, muchos de los cuales están desarrollados sobre plataformas propias, lo cual nos permite el proceso de enseñanza-aprendizaje, así como, de investigación en el desarrollo de mejoras al mismo por otros investigadores, de igual manera existen herramientas donde se han desarrollado de manera práctica modelos, tales como por ejemplo el modelo MOMOS (Materia Organica y MicroOrganismos del Suelo), el cual ha sido calibrado y validado con datos e información de un estudio en un gradiente altitudinal tropical [Pansu et al., 2010, 2014] mostrando buenos resultados en las diferentes evaluaciones, este mismo modelo ha sido evaluado, encontrando una propuesta de evolución del mismos (Valery, 2018). Sin embargo, las investigaciones antes mencionadas, presenta la desventaja de haber sido desarrolladas en un software de simulación privativo llamado Vensim creado por Ventana Systems, Inc. para la construcción de modelos de sistemas dinámicos, limitado a funciones desarrolladas por dicha empresa, lo que no permite ampliar la investigación en otros aspectos de interés sin la necesidad de acudir a otros softwares externos.

De igual manera, el grupo de Investigación en Biotecnología Agrícola y Ambiental (GIBAA) de la UNET, ha desarrollado diversos planteamientos de modelos bajo un software libre, por lo que se propone el uso de la herramienta para el análisis estadístico y gráfica llamado R, el cual es un ambiente de programación formado por un conjunto de herramientas muy flexibles que pueden ampliarse fácilmente mediante paquetes, librerías o definiendo nuestras propias funciones. Además, es gratuito y de código abierto. Cualquier usuario puede descargar y crear su código de manera gratuita, sin restricciones de uso, la única regla es que la distribución siempre sea libre (GPL). Gracias a que puede accederse libremente a su código, R no tiene limitadas sus funciones, al contrario de lo que sucede con otras herramientas comerciales.

La finalidad de esta investigación consiste en utilizar el modelo MOMOS desarrollado por Valery

(2018) en Vensim e implementarlo en R, permitiendo la comparación de los resultados obtenidos en Vensim (Valery, 2018) contra el resultado del paquete a desarrollar. Buscando como meta la versatilidad a los investigadores interesados de poder trabajar sobre un ambiente que puedan manejar y modificar en las diferentes necesidades de trabajo.

## 1.2. Objetivos

### 1.2.1. Objetivo General

Desarrollar un paquete en el lenguaje R que implemente el modelo MOMOS.

### 1.2.2. Objetivos Específicos

- Estudiar la Estructura del modelo MOMOS.
- Crear las funciones primarias en el lenguaje R para implementar el modelo MOMOS.
- Realizar las pruebas unitarias y funcionales del paquete creado en el lenguaje R.

## 1.3. Justificación e Importancia

En las ciencias agronómicas y áreas afines, en la última década, se han utilizado una serie de herramientas que permiten mejorar los procesos de producción, así como del manejo de los recursos, entre las mismas, se encuentran los modelos dinámicos de simulación los cuales lográn mostrar una representación lo más cercana a la realidad, generando un mejor entendimiento de los procesos en los ecosistemas y agroecosistemas.

## 1.4. Alcance y Limitaciones

El paquete a desarrollar en el lenguaje R permitirá simular el modelo MOMOS, el resultado obtenido se comparará con los obtenidos por Valery (2018) en el programa VENSIM para simular

el mismo modelo.

# Capítulo 2

# Fundamentos Teóricos

## 2.1. Antecedentes

Contreras (2018) construyó: "DimBio, Paquete en lenguaje R para la discrminización de modelos de simulación dinámicos para procesos biológicos.", en este trabajo se expone de forma detallada la realización de un paquete en lenguaje R, para la discriminación de modelos de simulación dinámicos para procesos biológicos, para unificar los estadísticos que evaluan el desempeño de los modelos dinámicos, mediante el cálculo para discrimación de estadísticos univariantes y multivariantes utilizados para encontrar un modelo adecuado o que mejor se ajustará al proceso de estudio basado en modelos.

La investigación fue de relavancia para este trabajo de grado, ya que en ella crearon un paquete en R, y utilizaron modelos de simulación dinámicos.

Darghan (2018) realizó: "CGR, Paquete en lenguaje R, para el cálculo de índices fisiológicos de crecimiento y componentes del rendimiento en plantas." trabajo en el cual se expone el desarrollo del paquete CGR cuyo interés radicó en el cálculo de aquellos índices que dependen de la primera derivada de la forma funcional ajustada al modelo de crecimiento de las plantas, también permite el cálculo de índices instantáneos en cualquier punto de interés por el usuario dentro de la región experimental o dentro del dominio de la función ajustada.

La investigación citada fue tomada como referencia, puesto que realizó cálculos de índices para evaluar el comportamiento de ciertas variables en procesos biológicos, en este caso, el crecimiento. Lo cual está vinculado al contexto biológico de los datos que fueron evaluados por el paquete que

se desarrolló.

Ablan (2011) construyó: "Una librería en R para validación de modelos de simulación", en este trabajo se manejaron métodos para la validación de modelos de simulación continua para evaluar la calidad del modelo comparando los resultados de series temporales de datos reales con los resultados o salidas obtenidos de una simulación para el mismo lapso temporal, en el cual se utilizaron los criterios o índices de validación que pueden ser usados para la comparación de datos y resultados de un modelo, los cuales fueron: el error cuadrático medio, el estadísticos de Theil, el error absoluto medio, el índice de acuerdo de Willmott, la eficiencia del modelo de Nash Sutcliffe, el coeficiente de correlación de Pearson, el coeficiente de determinación, la prueba F, la prueba t pareada, el estadístico Hotelling T2, el criterio de información de Akaike y la inferencia bayesiana. Con estos criterios o índices se creó una librería del programa estadístico R que permitió el uso de estos índices para la validación de modelos de simulación.

La investigación fue importante en este trabajo de grado, ya que en ella crearon una librería de R, y utilizaron métodos estadísticos para respuesta univariante que son utilizados para la comparación de datos reales contra datos de modelos de simulación.

## 2.2. Bases Teóricas

### 2.2.1. Modelos

En la actualidad el avance de la tecnología, permite que para el estudio de sistemas biológicos, los modelos de simulación sean una herramienta que ayuda en el manejo de la información, así como la evaluación y planificación de manejos y diseño de experimentos.

Existe una gran cantidad de modelos biológicos, los cuales están relacionados con diversos procesos en los ecosistemas o agroecosistemas. Entre estos se pueden observar modelos teóricos, diagramáticos o matemáticos. "Estos modelos han venido evolucionando desde la última dácada, buscando entender los procesos que permiten el almacenamiento o la pérdida del suelo." (Manzoni, 2009).

El uso de modelos en el área ecológica puede estar enfocado en diversos objetivos, entre los que se pueden destacar: la representación de diversas variables de estudio y sus tasas de cambio;

la descripción de ecosistemas y los procesos temporales y/o espaciales de los mismos; evaluar las condiciones pasadas y predecir el comportamiento futuro de los ecosistemas en estudio; poner a prueba teorías o hipótesis sobre la estructura y el funcionamiento de los ecosistemas (Blanco, 2013).

Dentro del grupo de modelos que estudian las transformaciones de la materia orgánica del suelo (MOS), se presentan características similares entre ellos, tales como, estar conformados por compartimientos, que representan a los diferentes compuestos químicos o biológicos, tambien se pueden diferenciar, por el número de compartimientos que poseen, que pueden ir desde 2 hasta 200 compartimientos. Sin embargo, "debe tenerse en cuenta un equilibrio entre el detalle y la comprensión de la información, para no perder el sentido de la modelización y poder realizar simulaciones y predicciones acordes a la realidad" (Blagodatsky 1998; Bruun 2003; Kirschbaum 2002; Zhang 2008).

La literatura presenta el interés que existe por los modelos que tienen que ver con las transformaciones de la MOS, muestra de esto es la gran cantidad de modelos que son referenciados en el registro de modelos ecológicos (REM `http://ecobas.org`), donde se presenta la información de cerca de 684 modelos.

Según lo presentado por Haefner (2005), sobre la importancia de los modelos y su potencial para el estudio de sistemas naturales, se tiene que mediante los mismos se puede lograr:

1. Sintetizar, manejar y unificar una gran cantidad de información en forma de ecuaciones y sus relaciones, permitiendo entender y explicar el comportamiento de los sistema biológicos.
2. Analizar el funcionamiento del sistema para predecir el comportamiento futuro, bajo las condiciones naturales o con intervención de manejo.

"Los modelos de simulación plantean un esquema básico de trabajo, el cual es fundamentado en tres etapas generales" (Haefner, 2005; Liu 2005), las cuales pueden ser descritas de la siguiente manera:

1.- Conceptualización. Conocer en gran medida la realidad que se trata de modelizar, ser capaces de representar lógicamente y de manera conceptual el problema.

2.- Formalización. establecer correctamente las relaciones entre los elementos que conforman el sistema en estudio, de forma que sean comprensibles, lo que permite construir un diagrama de

estas relaciones y posteriormente representarlo mediante un diagrama de *Forrester*.

3.- Evaluación. Establecer la forma en que debe ser el procedimiento de resolución a emplear, y la manera de interpretarlo correctamente.

Las etapas de conceptualización y formalización se basan en toda la información de décadas pasadas, sin embargo, el proceso de evaluación presenta diversos problemas, entre los que destacan la recolección de información con características confiables que sea representativa de los sistemas y validar los resultados obtenidos por el modelo con los obtenidos experimentalmente mediante un análisis estadístico inferencial.

### 2.2.2. Lenguaje R.

R es un conjunto integrado de *software* de código abierto para el almacenamiento, manipulación, cálculo y visualización de datos para computación y graficación estadística, puede ser compilado y ejecutado en 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 comúnmente 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 éxitoso, 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). el desarrollo del algoritmo del modelo MOMOS se realiza empleando R y la herramienta Rstudio.

### 2.2.3. 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). También es necesario conocer con precisión que tipo de lenguaje es y una buena forma de realizar el algoritmo que simula el modelo MOMOS.

### 2.2.4. Algoritmos en R/RStudio.

El lenguaje R es un lenguaje interpretativo y no compilado, esto se traduce a que los comandos escritos por teclado son ejecutados directamente sin necesidad de construir un ejecutable, lo cual representa una facilidad al momento de analizar datos complejos.

Como lenguaje de programación, R adopta una sintaxis y una gramática que difiere de muchos otros lenguajes: los objetos en R son vectores, y las funciones son vectorizádas para operar con todos los elementos del objeto. Los objetos en R tienen una semántica de copia sobre el cambio y de paso por valor, reduciendo las consecuencias inesperadas para los usuarios en detrimento del uso menos eficiente de la memoria. Es importante puntualizar que los paradigmas comunes en otros lenguajes, como el bucle 'for', no son tan comunes en R. Los programas escritos en R reflejan de una manera más clara los algoritmos estadísticos que implementan. Además, cuenta con la ventaja de que es fácil de depurar código en R por ser un lenguaje interpretado. (Becerro, 2014).

Para poder comparar el modelo MOMOS experimental versus el modelo simulado, es necesario leer los datos experimentales, como método de obtención será empleado la lectura de datos desde un archivo de formato ".xls". Los datos leidos son aquellos datos experimentales del modelo que fueron capturados en un momento determinado y que escapa del contexto de este estudio.

**xlsx:** es un paquete para R que da control programático de archivos excel. Este paquete emplea una API de alto nivel que permite al usuario leer de una hoja de un archivo .xlsx a un data.frame y escribir un data.frame a un archivo. Funcionalidad de menor nivel permite la manipulación directa de hojas, filas y celdas. Para poder visualizar el contenido de agua relativo por capa de suelo de manera gráfica (resultante de la simulación), es necesario emplear las herramientas provistas por

el paquete de R descrito a continuación.

**ggplot2:** es un paquete para R que permite crear elegantes visualizaciones de datos usando gramática de descripción de gráficos, es un sistema para declarativamente crear gráficos. Una vez la data ha sido provista, mediante funciones se le indica a este paquete como graficarlas.

### 2.2.5. Estructura de paquetes en R/RStudio.

La estructura de un paquete en R cuenta con al menos los cuatro ítems siguientes:

1. **DESCRIPTION:** En este componente se encuentra la metadata del paquete. La tarea del archivo Description es de gran importancia, ya que es en 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 DESCRIPTION de 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 (¿= 3.1.0)
   - License: What license is it under?
   - LazyData: true

2. **R/Directorio:** Dirección del repositorio donde se encuentra el código del paquete, es decir los “.R” archivos. Se exponen las buenas prácticas a la hora de realizar todo el código en R, desde organización de las funciones, estilos de código, comentarios y nombre de variables.

   - Organizar funciones en R: Aunque se puede organizar los archivos como se desee, los dos extremos son malos, no colocar todas las funciones en el mismo archivo y no crear

un archivo para cada 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 significado y deben de terminar en R. Se puede recomendar de acuerdo al número de funciones utilizar prefijo.

- Nombres de objetos: Los nombres de las variables y funciones deben de ser en minúsculas, usar el guión bajo ( _ ) para separar palabras. En lo posible no debe usarse nombres de variables existentes, esto causará confusión.
- Comentarios: Para comentar el codigó, se comienza con #, los comentarios deben de explicar el porqué, no el que. Se usan los caracteres (-) y (=) para separar líneas.

- Estilos de código: Existe diferencia entre el código utilizado en *scripts* y paquetes:
  en un *script*, el código se ejecuta cuando se carga, mientras en un paquete el código se ejecuta cuando se genera. Esto significa que el código de paquetes solo debe crear objetos, a continuación se amplíam estas importantes diferencias:
  - Cargando código: Cuando se carga un *script* el código se ejecuta de una vez, el procedimiento es diferente en un paquete, porque es cargado en dos pasos, el primero, cuando se construye el paquete todo el código se ejecuta en R/ y su resultado es guardado, el otro paso es cuando se carga un paquete, con *library()* o *require()*, los resultados almacenados en la caché están dispuestos para su uso.
  - The R landscape: Hay también diferencia en un *script* y un paquete, por lo tanto hay que prestar atención a R *landscape*, incluyendo las funciones, objetos y todas las variables globales.

Recomendaciones:

- No se suele utilizar library() o require(). Estos modifican la ruta de búsqueda, lo que afecta las funciones disponibles del entorno global. Es mejor usar la descripción para especificar los requisitos de un paquete.
- Nunca emplear *source()* para cargar el código de un archivo. Ya que *source()* modifica el entorno actual, insertando los resultados de ejecutar el código. En cambio, usando el devtools::load_all() automáticamente se generarán todos los archivos en R.

3. **Man/Documentación:** La documentación es uno de los aspectos más importantes de un buen paquete. Sin ella, los usuarios no sabrán cómo usar un paquete y además ayudan a recordar que realizan sus funciones para el futuro.

R proporciona una forma estándar de documentar los objetos en un paquete: Se escriben archivos .Rd en el directorio "man". Éstos tilizan LaTeX, y se procesan en HTML, texto plano y pdf para su visualización. En lugar de escribir estos archivos a mano, se emplea "roxygen2", que convierte los comentarios especialmente formateados en archivos ".Rd", el objetivo de roxygen2 es hacer que documentar el código sea lo más fácil posible.

Los comentarios roxgen utilizan # si se utilizan para distinguir de los comentarios regulares.

Para documentar funciones se utilizan generalmete 3 etiquetas que son:

- @param: donde se describe los parámetros de entrada de la función.
- @examples: donde se muestra un ejemplo funcional de la función ya que R CMD chequea que sea correcto.
- @return: específica lo que retorna la función siempre que sea necesario.

Para documentar el paquete se recomienda utilizar roxygen con la etiquetas @docType package y @name (nombre del paquete) también se puede utiliza la etiqueta @section que permite ser específicos al momento de documentar.

Documentación de clases, genéricos y métodos:

- S3: La clase S3 no tiene definición formal así que documenta la función constructora. No se necesita documentar métodos simples pero si el método es complicado o incluye argumentos adicionales se debe de documentar.
- S4: La clase S4 utiliza roxygen para realizar la documentación, los S4 también son funciones así que se deben documentar como tal, la diferencia con S3 es que todos los métodos deben documentarse. A menudo el código S4 necesita ejecutarse en cierto orden, por ejemplo, para definir los métodos "setMethod("foo", c("bar","baz"), ...)" se deben haber creado antes el "foo" genérico y las 2 clases.
- RC: Las clases RC de referencia son diferentes a S3 y S4 porque los métodos están asociados con clases, no con genéricos. La "docstring" es una cadena colocada dentro de la definición del método que describe brevemente lo que hace. Esto hace que documentar RC sea más simple que S4 porque solo se necesita un bloque de roxygen por clase.

4. **NAMESPACE:** Específica que objetos conforman el paquete. Se utiliza para proporcionar espacios a los nombres como su nombre lo indica, es utilizado para interactuar con los paquetes y sus variables del CRAN de R.

### 2.2.6. Pruebas estándar y pruebas funcionales

Para poder evaluar si el algoritmo y su componentes satisfacen o no los requerimientos para el modelos MOMOS, es necesario realizar un proceso de pruebas. Una prueba es definída según Tutorialspoint (Software testing, sf.) como *.A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item".*

En el contexto de este estudio se consideraran las pruebas estándar como el conjunto de pruebas de nivel que comprenden las pruebas unitarias, las pruebas de sistema y las pruebas de aceptación.

- **Pruebas Unitarias:** "Unit testing is performed by the respective developers on the individual units of source code assigned areas. The goal of unit testing is to isolate each part of the pro- gram and show that individual parts are correct in terms of requirements and functionality". La data para efectuar estas pruebas suele ser diferente de la que es empleada en las pruebas de aceptación.

- **Pruebas de Sistemas**: "System testing tests the system as a whole. Once all the components are integrated, the application as a whole is tested rigorously to see that it meets the specified Quality Standards",(Tutorialspoint, Software testing, sf). Las pruebas de sistema permiten probar verificar y validar los requerimientos del sistema en conjunto con la arquitectura la aplicación.

- **Pruebas de Aceptación**: "Acceptance tests are (...) to point out simple spelling mistakes, cosmetic errors, or interface gaps, (and) to point out any bugs in the application that will result in system crashes or major errors in the application", (Tutorialspoint, Software testing, sf). Estas pruebas son para probar como la aplicación se comporta en producción.

Existe un tipo de prueba especial que és la que se va aplicar al final del desarrollo del algoritmo, estas son las pruebas funcionales. Tutorialspoint(Software testing, sf.) define las pruebas funcionales como:

*"This is a type of (...) testing that is based on the specifications of the software that is to be tested. The application is tested by providing input and then the results are examined that need to conform to the functionality it was intended for. Functional testing of a software is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements".*

### 2.2.7. Glosario

**Código fuente:** Conjunto de instrucciones que componen un programa, escrito en cualquier lenguaje. En inglés "source code".

**Crecimiento:** Aumento en un atributo medido, $x$, de un órgano u organismo, como una función del tiempo, $t; x = f(t)$.

**Data:** El nombre genérico para cualquier cosa que entre, salga o se guarde en una computadora o cualquier otro medio, siempre y cuando sea todo en formato digital.

**Modelo:** ≪Un modelo (M) para un sistema (S) y un experimento (E) es cualquier objeto al que puede aplicarse E para responder preguntas acerca de S≫ Marvin Minski (1965). Paul Humphrey define simulación como "Simulación es cualquier método implementado en computadora destinado a explorar las propiedades de modelos matemáticos, cuando no se dispone de métodos analíticos" (Grüne-Yanoff y Weirich 2010).

**Lenguaje R:** R es un entorno y lenguaje de programación con un enfoque al análisis estadístico.

**R Studio:** R Studio es un entorno de desarrollo integrado, en inglés "integrated development environment"(IDE) para R.

## 2.3. *GNU General Public License v2.0 (GPL-2.0)*

Es una licencia de *software* libre, que garantiza se pueda copiar, distribuir y modificar el *software* siempre que realice un seguimiento de los cambios / fechas en los archivos de origen. Cualquier modificación o software que incluya (a través del compilador) el código con licencia GPL también debe estar disponible bajo la GPL junto con las instrucciones de compilación e instalación.

# Capítulo 3

# Fundamentos Metodológicos

A continuación se plantea la metodología a para el presente trabajo, detallando el enfoque, tipo, nivel y diseño de la investigación y la metodología a implementar.

## 3.1. Enfoque de la investigación

La presente investigación se desarrollaró 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 módelos matemáticos y estadísticos, el conocimiento está basado en hechos". Los datos a usados en el desarrollo del paquete y para la comparacion de los resultados, provienen de la tesis Doctoral de Valery (2018).

## 3.2. Tipo o nivel de investigación

Este proyecto planteó un tipo de investigación de desarrollo, en donde se integra la programación y evaluación del comportamiento de un paquete de simulación, que permite indagar los efectos de la interrelación entre los diferentes tipos de variable en lugar de los hechos.

En este punto se determinó 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 midieron los datos extraídos sin alterarlos para ser mostrados en el sistema.

## 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 haber establecido una correcta secuencia de pasos para la elaboración del prototipo de software que dio solución a la problemática principal de la investigación.

Con este enfoque, se tiene que este trabajo siguió 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 creó un *software* aplicable al área de estudio.

## 3.4. Metodología

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

**Creación del esqueleto del paquete.**

En esta etapa se diseñaron y crearon los directorios, ficheros y objetos que conforman el paquete.

**Registrar el método para el envío y uso de funciones.**

En esta etapa del desarrollo se estableció 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 estableció los espacios de nombre o las estrategias para la búsqueda y utilización de las variables; unificando estos criterios a las funciones que fueron diseñadas.

**Diseño y codificación de las funciones.**

Los métodos para el diseño de las funciones primarias en R fueron los diagramas de flujo; y para su codificación se siguieron las normas de estilo para codificación en R, sugeridas por Wickham (2015) y por el creador del paquete *formatR* Xie(2017), además se estableció 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 realizaron pruebas unitarias en dos fases, la primera con datos sintéticos que permiten comprobar cada estado del diagrama de flujo, que esquematiza la solución numérica que permite el cálculo, para esto se utilizó el paquete de R*RUnit* (Zenka, 2015) y *testthat* (Wickham, 2017), y la segunda etapa donde el modelo final ya implementado permite realizar pruebas con los conjuntos de datos facilitados que formaron parte integral del paquete y con los cuales se desarrollaron los ejemplos prácticos que conformaron la documentación que acompaã al paquete R.

**Chequear la carga del paquete.**

En esta etapa del desarrollo se utilizó las funciones de chequear el 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 crea el archivo de documentación en LaTeX y/o HTML, compila el código fuente y crea las librerías de enlace dinámico (*dynamic link library* DLL).

**Construcción del método de distribución del paquete.**

Se seleccionó 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 fue planificada según lo establecido en el siguente diagrama:

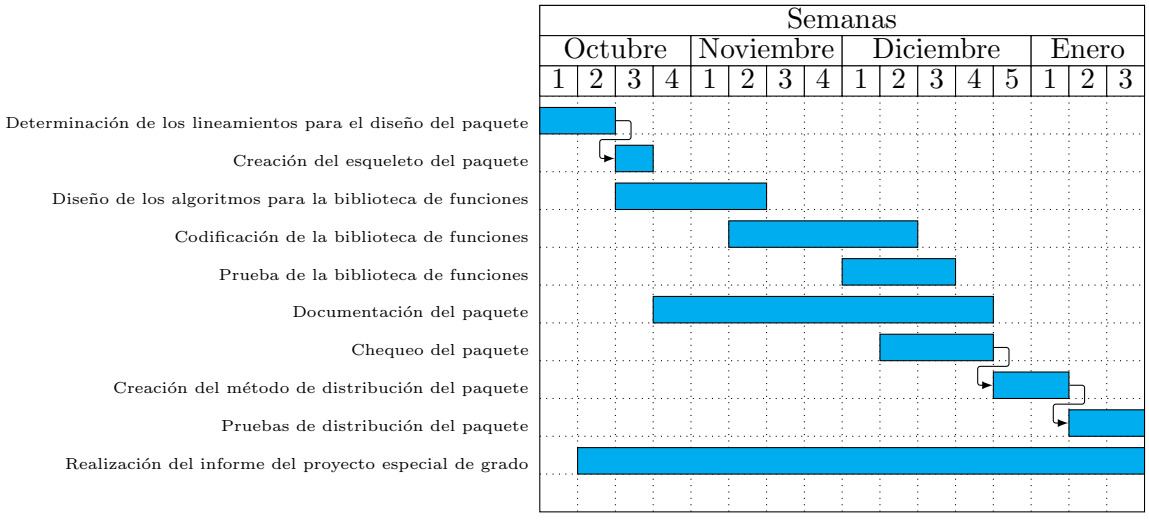


Figura 3.1: Diagrama de Gantt con la planficación del proyecto especial de grado

# Referencias Bibliográficas


Martínez Rodríguez, Francisco y Otros: Lombricultura. Manual práctico, Impreso: Unidad de Producciones Gráficas MINREX, La Habana, Cuba, 2003.

Rujano M., Ablan M., Sarmiento L. (2011). Simulación de la respuesta de la materia orgánica del suelo en diferentes ecosistemas ante escenarios de cambio climático en Venezuela. Disponible en: http://erevistas.saber.ula.ve/index.php/ecodiseno/article/view/4372/4149. Consultado Septiembre 2019.

Universidad Complutense Madrid. Conservación de los recursos naturales para una Agricultura sostenible: Materia orgánica y actividad biológica. Disponible en: https://www.ucm.es/data/cont/media/www/pag-104576/1.%20Materia%20org%C3%A1nica%20y%20actividad%20biol%C3%B3gica.pdf. Consultado Septiembre 2019.

Tuomi, M., Vanhala, P., Karhu, K., Fritze, H., and Liski, J. (2008). Heterotrophic soil respiration-comparison of different models describing its temperature dependence. Ecol.Model., 211:182–190.

Valery A.(20xx). La temperatura y humedad como reguladores de la descomposición de la MOS: desempeño de diversas funciones de respuesta en un gradiente altitudinal tropical.

Ferrero R. (2018). Qué es R Software. Disponible en: https://www.maximaformacion.es/blog-dat/que-es-r-software/. Consultado Septiembre 2019.

Darghan K. (2018). CGR Paquete en lenguaje R, para el cálculo de índices fisiológicos de crecimiento y componentes del rendimiento en plantas. (Trabajo Especial de Grado de pregrado). Universidad Nacional Experimental del Táchira. San Cristóbal, Estado Táchira.

Contreras R.(2018). DiMBio, paquete en lenguaje R para la discriminación de modelos de simulación dinámicos para procesos biológicos. (Trabajo Especial de Grado de pregrado). Universidad Nacional Experimental del Táchira. San Cristóbal, Estado Táchira.