

	<b>Guía de estilo para la programación en R</b>	<b>CÓDIGO: G6_DE</b>
		<b>VERSIÓN: 1</b>
	<b>Direccionamiento Estratégico</b>	<b>FECHA DE APROBACION: 13/12/2024</b>
		<b>Página: 1 de 6</b>

## 1. OBJETIVO

Generar código en lenguaje de programación R con un estilo uniforme y coherente, que facilite su lectura y comprensión y garantice la transparencia, auditabilidad, mantenimiento, implementación y colaboración en los procesos que soporta.

## 2. ALCANCE

Este documento establece las convenciones generales para la programación y generación de código en el lenguaje de programación R, está dirigido a los profesionales de los diferentes grupos de trabajo de la Agencia Distrital para la Educación Superior, la Ciencia y la Tecnología – ATENEA, que desarrollen productos de procesamiento y análisis de información en R.

Las convenciones aquí descritas se adaptan y adoptan de diferentes guías de estilo y son propias de la sintaxis del lenguaje R, el cuál es el lenguaje predominante en los proyectos de procesamiento y análisis de datos de la agencia ATENEA.

Las convenciones se estructuran en los aspectos de; Nomenclatura, Sintaxis y Organización, cada uno de los cuales ocupa una sección del presente documento como se presentan a continuación.

## 3. NORMATIVIDAD ASOCIADA

- Decreto de Creación de la Agencia: 273 del 2020
- Ley 1581 de 2012. Régimen General de Protección de Datos Personales y sus decretos reglamentarios
- Ley 1712 de 2014. Ley de Transparencia y Acceso a la Información Pública.
- Ley 1266 de 2008, Por la cual se dictan las disposiciones generales del hábeas data y se regula el manejo de la información contenida en bases de datos personales, en especial la financiera, crediticia, comercial, de servicios y la proveniente de terceros países y se dictan otras disposiciones.

	<b>Guía de estilo para la programación en R</b>	<b>CÓDIGO: G6_DE</b>
		<b>VERSIÓN: 1</b>
	<b>Direccionamiento Estratégico</b>	<b>FECHA DE APROBACION: 13/12/2024</b>
		<b>Página: 2 de 6</b>

## 4. DEFINICIONES

**Código Fuente en R:** Secuencia de caracteres que representa un conjunto de acciones que debe realizar un computador, el código sigue la sintaxis del lenguaje de programación R.

**R:** “Es un lenguaje y entorno para gráficos y computación estadística. Es un proyecto GNU similar al lenguaje y entorno S que fue desarrollado en Bell Laboratories (antes AT&T, ahora Lucent Technologies) por John Chambers y colegas.” <https://www.r-project.org/about.html>.

## 5. DESARROLLO

### 5.1. NOMENCLATURA

Este conjunto de directrices hace referencia a la forma de nombrar los archivos y objetos como archivos de datos y de código y objetos como funciones y variables, entre otros componentes de los códigos de procesamiento y análisis.

#### 5.1.1. NOMBRE DE ARCHIVOS

Las siguientes directrices se toman de la guía de estilo de código en R de Hadley Wickham (Hadley, s.f.), salvo que se indique lo contrario.

- Los nombres de los archivos de código deben ser significativos, es decir, deben representar fielmente el proceso que ejecutan, por ejemplo, si el código realiza el ordenamiento de los puntajes el nombre puede ser: “orden-puntaje.R”.
- Los nombres deben estar en idioma español y no deben contener tildes, “ñ” ni caracteres especiales (adaptación propia para el idioma español).
- No utilice letras mayúsculas.
- Asigne nombres cortos y auto explicativos.
- Si el nombre está compuesto por dos o más palabras sepárelas con un guion medio.
- Si el proyecto o proceso requiere ser ejecutado en una secuencia específica añada prefijos con numerales según el orden.
  - Ejemplo:
    - 0-cruce-habilitacion
    - 1-puntuacion
    - 2-ordenamiento

#### 5.1.2. NOMBRE DE OBJETOS

**Piensa en el medio ambiente, antes de imprimir este documento.**

Cualquier copia impresa de este documento se considera como COPIA NO CONTROLADA

	<b>Guía de estilo para la programación en R</b>	CÓDIGO: G6_DE
		VERSIÓN: 1
	<b>Direccionamiento Estratégico</b>	FECHA DE APROBACION: 13/12/2024
		Página: 3 de 6

Los siguientes lineamientos se toman de la guía de estilo de código en R de Hadley Wickham (Hadley, s.f.)

- Los nombres de variables y funciones deben escribirse en minúscula y separados por guion bajo, aplican las demás recomendaciones de nombres de archivos:
- Los nombres de variables deben ser sustantivos y los de funciones verbos en infinitivo ej: calcular puntajes
- Evite usar nombres de funciones que ya estén en uso
- Evite usar nombres similares para diferentes objetos
- Los nombres de las constantes siguen los lineamientos anteriores, pero con el prefijo k, ej:
- “k\_num\_cupos”

## 5.2. SINTAXIS

Esta sección describe las directrices sobre la estructura y forma de digitación del código, incluye directrices sobre el uso de los espacios, longitud de las líneas de código y uso de llaves.

Los siguientes lineamientos se toman de la guía de estilo de código en R de Hadley Wickham (Hadley, s.f.), salvo que se indique lo contrario.

### 5.2.1. ESPACIADO

- Antes y después de operadores (+,-,=) siempre debe haber un espacio
- Antes de una coma no debe haber espacio, después de una coma si debe haber espacio
- Antes del paréntesis izquierdo hay un espacio, excepto si se está llamando a una función
- Cuando se quiere ordenar el código, alineándolo, por ejemplo, al signo de igualdad como en el siguiente ejemplo, está bien agregar más espacios:
- `data.frame(`
- `nombres = c("Maria", "José", "John"),`
- `resultados = c(7, 3, 8),`
- `convocatoria = c(JU1, JU2, JU1)`
- Los paréntesis y corchetes no llevan espacio alrededor:
- Bien:
- `resultados[5, 1]`
- Mal:

**Piensa en el medio ambiente, antes de imprimir este documento.**

Cualquier copia impresa de este documento se considera como COPIA NO CONTROLADA

	<b>Guía de estilo para la programación en R</b>	<b>CÓDIGO: G6_DE</b>
		<b>VERSIÓN: 1</b>
	<b>Direccionamiento Estratégico</b>	<b>FECHA DE APROBACION: 13/12/2024</b>
		<b>Página: 4 de 6</b>

- resultados[ 5, 1 ]

### 5.2.2. LLAVES

- Las llaves de apertura no deben ir en su propia línea, las llaves de cierre deben ir en su propia línea ej:

```
if (x <0 && debug) {
  message("x es negativo")
}
```

### 5.2.3. ASIGNACIONES

- Para realizar asignaciones use el símbolo <-, en lugar de =
  - No se permiten asignaciones del lado derecho, tales como (Google, s.f.):  
dplyr::summarize(puntaje, max\_puntaje = max(puntaje)) -> resultados

### 5.2.4. LONGITUD DE LÍNEAS

Use máximo 80 caracteres por cada línea de código.

### 5.2.5. USO DE RETURN ()

Utilice explícitamente la función de retorno "return()" de manera explícita, para evitar conflictos no confíe en la función de retorno explícita de R. (Google, s.f.).

## 5.3. ORGANIZACIÓN

Esta sección establece directrices sobre cómo organizar el código, tanto a nivel de archivos de código como a nivel de líneas de código dentro de los archivos, específicamente se ofrecen directrices sobre el uso de comentarios, la separación de fragmentos de código y el modularidad en los archivos de código.

### 5.3.1. COMENTARIOS

Use comentarios en el código, cada línea de comentario debe empezar con el símbolo # y un solo espacio, los comentarios deben explicar por qué y no el qué, el qué lo dice el código.

### 5.3.2. SEPARACIÓN

Para separar y estructurar el código en pedazos use la convención de comentario seguido de guiones, ej:

```
# Desempate -----
```

**Piensa en el medio ambiente, antes de imprimir este documento.**

Cualquier copia impresa de este documento se considera como COPIA NO CONTROLADA

	<b>Guía de estilo para la programación en R</b>	CÓDIGO: G6_DE
		VERSIÓN: 1
	<b>Direccionamiento Estratégico</b>	FECHA DE APROBACION: 13/12/2024
		Página: 5 de 6

### 5.3.3. MODULARIDAD

- Estructure todo el proyecto en módulos en vez de una pila, de tal forma que se permita actualizar o modificar cada sección del proyecto sin afectar los módulos precedentes o posteriores.
- En la medida de lo posible parametrize los elementos del código (rutas, archivos, etc.), de tal manera que las actualizaciones requieran cambios por única vez en el parámetro en lugar de hacerlo cada vez que se llama el elemento en el código.
- Use los nombres originales de los archivos de entrada con el fin de identificarlos precisamente por su nombre fecha y versión y asígnelos a una variable o parámetro.

### 5.4. DOCUMENTOS DE REFERENCIA

Google. (s.f.). *Google's R Style Guide*. Obtenido de <https://google.github.io/styleguide/Rguide.html>

Hadley, W. (s.f.). *Advanced R*. Obtenido de Style Guide: <http://adv-r.had.co.nz/Style.html>

	<b>Guía de estilo para la programación en R</b>	CÓDIGO: G6_DE
		VERSIÓN: 1
	<b>Direccionamiento Estratégico</b>	FECHA DE APROBACION: 13/12/2024
		Página: 6 de 6

## 6. CONTROL DE CAMBIOS

Fecha	Versión	Descripción del Cambio

VALIDACIÓN	NOMBRE	CARGO	FECHA
<b>Elaboró</b>	Raúl Andrés Gómez Aldana	Profesional	18/11/2024
<b>Revisó</b>	John Alejandro Torres Sichacá	Profesional especializado	12/12/2024
<b>Aprobó</b>	Javier Andrés Rubio Sáenz	Sugerente Análisis de la información y Gestión del Conocimiento	13/12/2024
<b>Aprobó</b>	Ximena Pardo Peña	Sugerente de Planeación	13/12/2024

**Piensa en el medio ambiente, antes de imprimir este documento.**

Cualquier copia impresa de este documento se considera como COPIA NO CONTROLADA