

**UNIVERSIDAD TECNICA PARTICULAR DE LOJA**

**Facultad de Ingenierías y Arquitectura**

**Fundamentos de Análisis de Datos**

**AUTOR**

Oliver Roberto Saraguro Remache

**DOCENTE**

Ing. Eduardo Encalada

26 de mayo de 2025

**Descripción de la API seleccionada**

*Razones*

|  |
| --- |
| * Es gratuita y no requiere autenticación. * Proporciona información en tiempo real sobre criptomonedas. * Permite realizar múltiples consultas, desde precios actuales hasta criptos populares. |

*Propósito*

|  |
| --- |
| La API de CoinGecko está diseñada para brindar datos actualizados en tiempo real sobre el mercado de criptomonedas. Entre los tipos de datos que ofrece están:   * Precios actuales y variaciones. * Capitalización de mercado y volumen de trading. * Rankings de criptomonedas. * Datos históricos y gráficas (sparkline). * Información sobre exchanges, categorías, tokens populares, entre otros. |

*Estructura*

|  |
| --- |
| La API trabaja principalmente con el formato JSON, y los datos pueden obtenerse utilizando solicitudes HTTP del tipo GET. Los endpoints están organizados por función (mercado, historial, búsqueda, categorías, etc.) y permiten filtrar y ordenar la información mediante parámetros fácilmente configurables. |

**Requerimientos de datos realizados.**

**Requerimiento 1: Ranking de criptomonedas por capitalización**

*Descripción.*

|  |
| --- |
| Se obtiene un conjunto de 50 datos que muestre el ranking actualizado de las principales criptomonedas del mercado, ordenadas por su capitalización de mercado. Esta métrica es ampliamente utilizada para evaluar el valor total de una criptomoneda en circulación, lo que permite identificar cuáles son las monedas más relevantes y estables dentro del ecosistema cripto. |

*URL de la llamada a la API*

|  |
| --- |
| <https://api.coingecko.com/api/v3/coins/markets?vs_currency=usd&order=market_cap_desc&per_page=50&sparkline=true> |

*Explicación de los parámetros establecidos.*

|  |
| --- |
| * **vs\_currency**=usd: precios expresados en dólares. * **order**=market\_cap\_desc: ordena por capitalización de mercado (descendente). * **per\_page**=50: número de criptomonedas. * **sparkline**=true: incluye precios de los últimos 7 días. |

*Código R usado para la extracción y carga en el data frame.*

|  |
| --- |
| urlApi = "https://api.coingecko.com/api/v3/coins/markets"  moneda = "usd"  orden = "market\_cap\_dsc"  observaciones = 50  grafica\_7d = "true"  url = paste0(urlApi,  "?vs\_currency=", moneda,  "&order=", orden,  "&per\_page=", observaciones,  "&sparkline=", grafica\_7d)  rpta = GET(url)  class(rpta)  status\_code(rpta)  cntApi = content(rpta, "text", encoding = "UTF-8")  cnt = fromJSON(cntApi, flatten = TRUE) |

*Captura de pantalla del data frame, con los datos cargados*

|  |
| --- |
|  |

**Requerimiento 2: Criptomonedas populares hoy**

*Descripción.*

|  |
| --- |
| Este requerimiento permite conocer cuáles son las criptomonedas más populares del momento, es decir, las que más interés están generando entre los usuarios en plataformas como CoinGecko. Esto se basa en el comportamiento de búsqueda de los usuarios, no solamente en su valor de mercado. |

*URL de la llamada a la API*

|  |
| --- |
| <https://api.coingecko.com/api/v3/search/trending> |

*Explicación de los parámetros establecidos.*

|  |
| --- |
| La consulta al endpoint /search/trending no requiere parámetros adicionales, ya que devuelve directamente una lista predefinida de CoinGecko por criptomonedas más buscadas o en tendencia en las últimas horas en su plataforma. |

*Código R usado para la extracción y carga en el data frame.*

|  |
| --- |
| urlApi2 = "https://api.coingecko.com/api/v3/search/trending"  rpta2 <- GET(urlApi2)  class(rpta2)  status\_code(rpta2)  cntApi2 <- content(rpta2, "text", encoding = "UTF-8")  cnt2 <- fromJSON(cntApi2, flatten = TRUE)  cnt2 = cnt2[[1]] |

*Captura de pantalla del data frame, con los datos cargados*

|  |
| --- |
|  |

**Tareas de limpieza, transformación, y enriquecimiento realizados**

*Dataset seleccionado*

|  |
| --- |
| El dataset del primer requerimiento (cnt). |

*Lista de tareas de limpieza, transformación, y enriquecimiento realizadas*

|  |
| --- |
| Limpieza:   * Renombrar variables a español comprensible. * Eliminar la "T" y "Z" en el campo last\_updated. * Transformar fechas al tipo POSIXct.   Transformación:   * Seleccionar solo columnas relevantes. * Redondear variables numéricas a 2 decimales.   Enriquecimiento:   * Calcular precio máximo, mínimo y promedio de los últimos 7 días (variable sparkline\_in\_7d.price). |

*Código R usado para dichas tareas*

|  |
| --- |
| # Reducir dimencionalidad (TRANFORMACION)  cnt = cnt %>%  select(symbol, name, image, current\_price, market\_cap, market\_cap\_rank,  high\_24h, low\_24h, price\_change\_24h, circulating\_supply, last\_updated,  sparkline\_in\_7d.price)  # Renombrar variables (LIMPIEZA)  cnt = cnt %>%  rename(simbolo = symbol,  nombre = name,  imagen = image,  precioActual = current\_price,  capitalizacion = market\_cap,  rankingCapitalizacion = market\_cap\_rank,  max24H = high\_24h,  min24H = low\_24h,  variacion24H = price\_change\_24h,  monedasCirculacion = circulating\_supply,  fechaActualizacion = last\_updated,  precios7D = sparkline\_in\_7d.price)  # ENRIQUECIMIENTO DE DATOS  # Agregar el valor max, min, promedio de la semana  cnt = cnt %>%  mutate(max7d = sapply(precios7D, max),  min7d = sapply(precios7D, min),  promedio7d = sapply(precios7D, mean))%>%  select(-precios7D)  # LIMPIEZA  cnt = cnt %>%  mutate(fechaActualizacion = gsub("T", "", fechaActualizacion),  fechaActualizacion = gsub("Z", "", fechaActualizacion),  fechaActualizacion = as.POSIXct(fechaActualizacion, format = "%Y-%m-%d %H:%M:%S"))  # TRANSFORMACION  cnt = cnt %>%  mutate(  precioActual = round(precioActual, 2),  capitalizacion = round(capitalizacion, 2),  max24H = round(max24H, 2),  min24H = round(min24H, 2),  variacion24H = round(variacion24H, 2),  monedasCirculacion = round(monedasCirculacion, 2),  max7d = round(max7d, 2),  min7d = round(min7d, 2),  promedio7d = round(promedio7d, 2)  ) |

Captura de pantalla del data frame final con datos preprocesados

|  |
| --- |
|  |