

# **Tema II - Importación y Limpieza de Datos para Fortificación de Alimentos**

## **Desafío 2: Zinc – Puertas a los Datos.**

Dr. Maicel Monzón Pérez

2025-03-24

### **Índice**

<b>El Zinc, un Micronutriente Esencial</b>	<b>1</b>
<b>Objetivo de la práctica</b>	<b>2</b>
<b>Instrucciones</b>	<b>2</b>
<b>Tarea 1: Importar un archivo CSV con Problema por valores faltantes</b>	<b>3</b>
<b>Tarea 2: Leer un archivo Excel con metadatos</b>	<b>4</b>
<b>Tarea 3: Parsear números con formato decimal europeo</b>	<b>4</b>
<b>Tarea 4: Convertir fechas en formato “dd/mm/aa”</b>	<b>5</b>
<b>Tarea 5: Forzar tipos de columna con <code>col_types**</code></b>	<b>5</b>
<b>Conclusión</b>	<b>6</b>
<b>Desafío completado: Zinc – Puertas a los Datos</b>	<b>6</b>

### **El Zinc, un Micronutriente Esencial**

El zinc es un mineral esencial para el cuerpo humano, desempeñando funciones críticas como:

- **Función Inmunológica:** Contribuye al funcionamiento normal del sistema inmunológico, ayudando al cuerpo a defenderse contra infecciones.

- Crecimiento y Desarrollo: Es necesario para el crecimiento, la división celular y la síntesis de proteínas, siendo crucial durante la infancia, la adolescencia y el embarazo.
- Salud Reproductiva: Juega un papel importante en la fertilidad masculina y la salud reproductiva tanto en hombres como en mujeres.
- Metabolismo: Participa en numerosas reacciones enzimáticas y en el metabolismo de carbohidratos, proteínas y grasas.

En este contexto, la fortificación alimentaria (como la de harinas, cereales o suplementos) es una estrategia clave para combatir la deficiencia en poblaciones vulnerables.

Así como el zinc es un micronutriente esencial que actúa como una “puerta de entrada” para diversos procesos biológicos en el cuerpo humano, la importación de datos es la “puerta de entrada” esencial para el análisis de información nutricional en R.

---

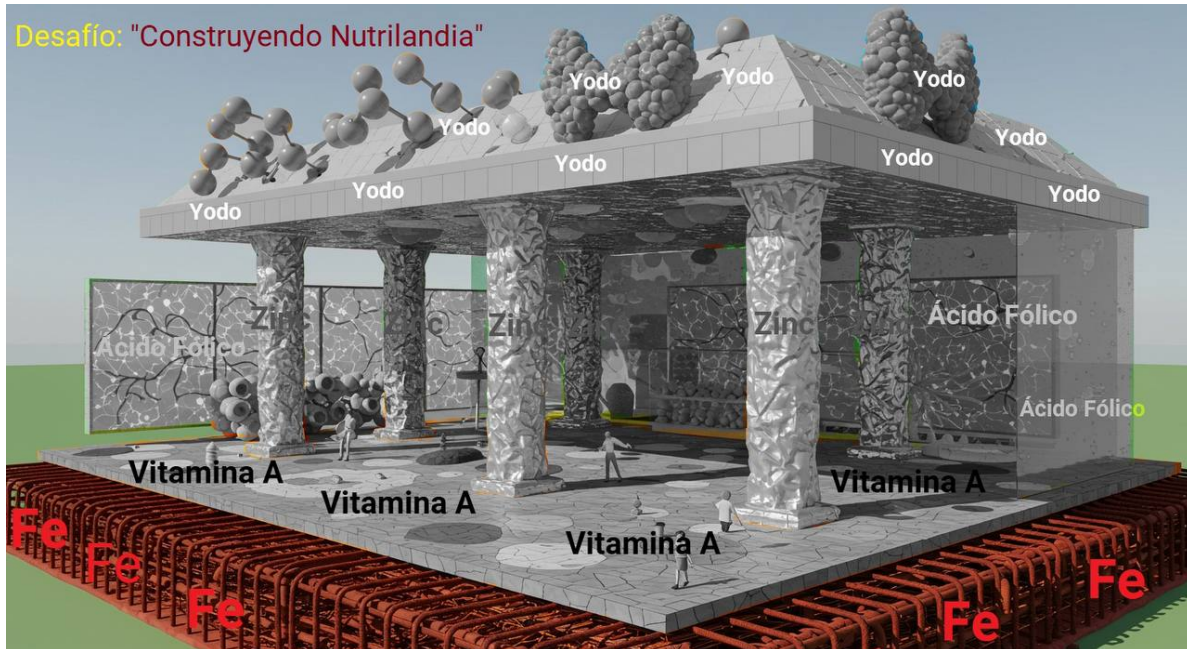
## Objetivo de la práctica

Convertirte en un “**ingeniero en leer ficheros**” al aprender a importar datos para realizar análisis de datos. Se un experto en el desarrollo de estructuras estables en el edificio de la fortificación de alimentos.

---

## Instrucciones

- Completa todas las tareas para obtener **5**.
- Usa el formato **R Markdown** o R script para entregar tus respuestas.



## Tarea 1: Importar un archivo CSV con Problema por valores faltantes

El archivo `datos_fortificacion.csv` contiene datos sobre la fortificación de alimentos, pero usa "-" para valores faltantes en la variable `hierro_mg`.

1. Importa el archivo usando `read_csv()`.
2. Asegúrate de que "-" se interprete como NA.

**Solución:**

```
library(readr)
datos_fortificacion <- read_csv("static/csv/datos_fortificacion.csv", na = c("-", "NA"))
# Verificar que los valores "-" están como NA en zinc_mg
# glimpse(datos_fortificacion)
```

---

## Tarea 2: Leer un archivo Excel con metadatos

El archivo `datos_fortificacion_metadatos.csv` tiene 2 filas de metadatos al inicio. Las columnas son: Año, País, Vitamina, Porcentaje.

1. Importa el archivo omitiendo las 2 primeras filas.
2. Asigna nombres de columnas manualmente.

**Solución:**

```
library(readr)
library(readr)
datos_fortificacion_metadatos <- read_csv("static/csv/datos_fortificacion_metadatos.csv",
  skip = 2)
# glimpse(datos_fortificacion_metadatos)
```

---

## Tarea 3: Parsear números con formato decimal europeo

Los datos en `datos_fortificacion_metadatos.csv` tienen valores como "12,5" (usando coma como decimal) .

1. Importa el archivo usando `read_csv()` .

**TIP:**

- Use la función `glimpse()` con sus datos como argumento para verificar el tipo de dato correcto

**Solución:**

```
library(readr)
datos_fortificacion_comas <- read_csv("static/csv/datos_fortificacion_comas.csv",
  locale = locale(decimal_mark = ","))
#view(datos_fortificacion_comas)
```

---

## Tarea 4: Convertir fechas en formato “dd/mm/aa”

La columna `fecha_encuesta` en `datos_fortificacion` está en formato cadena "1/1/2022". Parsea la columna como fecha usando `parse_date()` para llevar al tipo de datos correcto.

**TIP:** - Busca ayuda para entender la función `parse_date` Ej. “?parse\_date” - Use la función `glimpse` para verificar el tipo de dato correcto - La modificación solo se debe realizar una vez

**Solución:**

```
datos_fortificacion$fecha_encuesta <- parse_date(  
  datos_fortificacion$fecha_encuesta,  
  format = "%m/%d/%Y")  
# glimpse(datos_fortificacion) # Verificar tipo de dato "Date"
```

---

## Tarea 5: Forzar tipos de columna con `col_types**`

La columna `ingreso_mensual` se importó como `character` por un error en la base de datos `datos_fortificacion_metadatos.csv`.

1. Volver a importar el archivo forzando `ingreso_mensual` como numérico.

**Solución:**

```
library(readr)  
datos_fortificacion_metadatos <- read_csv("static/csv/datos_fortificacion_metadatos.csv",  
  col_types = cols(  
    ingreso_mensual = col_number(),  
    skip = 2)  
# view(datos_fortificacion_metadatos) # Verificar tipo numérico
```

---

## Conclusión

- **Resumen:**
  - Importar datos con `readr` y `readxl`, ajustando formatos y valores faltantes.
  - Transformar datos con `tidyr` para análisis estructurado.
  - Solucionar errores mediante `problems()` y `col_types()`.
- **Recomendaciones:**
  - Practicar con datos reales (ej. FAO o UNICEF).
  - Usar `janitor::clean_names()` para nombres de columnas uniformes.

## Desafío completado: Zinc – Puertas a los Datos

