

¡Hola! Vamos a analizar si las funciones en Excel (especialmente en su versión moderna, como Excel 365) pueden considerarse "ciudadanos de primera clase" (first-class citizens) en el contexto de la programación, y cómo esto se relaciona con las capacidades actuales de Excel, incluyendo las funciones como LAMBDA , LET , MAP , y otras.

¿Qué significa "ciudadano de primera clase" en programación?

En programación, un "ciudadano de primera clase" (first-class citizen) se refiere a una entidad (como una función, variable u objeto) que puede:

- Ser asignada a una variable.
- Pasarse como argumento a otras funciones.
- Devolverse como resultado de una función.
- Crearse dinámicamente durante la ejecución.

En lenguajes como JavaScript, Python o Java, las funciones son ciudadanos de primera clase porque pueden almacenarse en variables, pasarse como argumentos (por ejemplo, como callbacks), devolverse de otras funciones, o crearse en tiempo de ejecución (por ejemplo, con funciones anónimas o closures).

¿Son las funciones en Excel ciudadanos de primera clase?

Tradicionalmente, las funciones integradas de Excel (como SUM , IF , VLOOKUP , etc.) no eran ciudadanos de primera clase, ya que:

- No podían asignarse a variables ni almacenarse directamente.
- No podían pasarse como argumentos a otras funciones ni devolverse como resultados.
- Estaban limitadas a ser invocadas en celdas o fórmulas predefinidas.

Sin embargo, con la introducción de las funciones dinámicas y las capacidades modernas en Excel 365 (disponibles desde 2020), especialmente con la adición de LAMBDA , la situación ha cambiado significativamente. Vamos a evaluar esto paso a paso:

1. Asignación a una variable

- En Excel 365, puedes definir una LAMBDA y asignarla a un nombre (usando el administrador de nombres o la función NAME en algunos contextos). Por ejemplo:

excel

Contraer Ajuste Copiar

```
=DEFINE.NAME("Doble", LAMBDA(x, x * 2))
```

- Una vez nombrada, Doble actúa como una función personalizada que puedes usar en cualquier celda, como =Doble(A1) . Esto equivale a asignar una función a una variable en otros lenguajes.
- **Conclusión:** Sí, las LAMBDA pueden asignarse a nombres, lo que las hace similares a variables que contienen funciones.

2. Pasarse como argumento a otras funciones

- Las funciones dinámicas como MAP , BYROW , REDUCE , y FILTER permiten pasar una LAMBDA como argumento. Por ejemplo:

excel

Contraer Ajuste Copiar

```
=MAP(A1:A10, LAMBDA(x, x * 2))
```

- Aquí, la LAMBDA (LAMBDA(x, x * 2)) se pasa como un argumento a MAP , que la aplica a cada elemento del rango A1:A10 .
- Esto es análogo a pasar una función como callback en JavaScript (por ejemplo, array.map(x => x * 2)).
- **Conclusión:** Sí, las LAMBDA pueden pasarse como argumentos, un rasgo clave de ciudadanos de primera clase.

3. Devolverse como resultado de una función

- Excel no permite devolver una LAMBDA directamente como resultado de una fórmula en una celda (por ejemplo, no puedes escribir =LAMBDA(x, x * 2) en una celda y usarla como función). Sin embargo, puedes devolver el resultado de aplicar una LAMBDA dentro de otra función, y las LAMBDA nombradas persisten como funciones reutilizables.

- Ejemplo:

excel

Contraer Ajuste Copiar

```
=LET(
    func, LAMBDA(x, x * 2),
    func(5)
)
```

- Aquí, la LAMBDA se define dentro de LET y se aplica, pero no se devuelve como una función independiente. Para reutilizarla, debe nombrarse previamente.
- **Limitación:** No hay un mecanismo nativo para devolver una función como valor de celda que luego se pueda invocar (como en lenguajes funcionales puros). Esto es una restricción significativa.
- **Conclusión:** Parcialmente, no del todo. Las LAMBDA no se devuelven como funciones invocables, pero su lógica puede encapsularse y aplicarse dentro de otras fórmulas.

4. Creación dinámica durante la ejecución

- Las LAMBDA pueden crearse dinámicamente dentro de una fórmula en tiempo de ejecución. Por ejemplo:

excel

Contraer Ajuste Copiar

```
=MAP(A1:A10, LAMBDA(x, IF(x>0, x*2, x)))
```

- La LAMBDA se genera y aplica al rango en el momento de calcular la fórmula.
- Esto es similar a crear funciones anónimas en otros lenguajes.
- **Conclusión:** Sí, las LAMBDA pueden crearse dinámicamente, cumpliendo con este criterio.

Evaluación general

Con las funciones dinámicas y LAMBDA en Excel 365, las funciones (específicamente las LAMBDA) cumplen parcialmente los criterios de ciudadanos de primera clase:

- **Sí:** Pueden asignarse a nombres, pasarse como argumentos a funciones como MAP o BYROW, y crearse dinámicamente.
- **No completamente:** No pueden devolverse como funciones invocables desde una celda ni persistir como objetos independientes fuera de un nombre predefinido.

Comparado con lenguajes de programación puros (como JavaScript, donde una función puede asignarse a una variable y devolverse como `return function() { ... }`), Excel tiene limitaciones debido a su naturaleza de hoja de cálculo, donde las celdas contienen valores o fórmulas, no objetos programáticos complejos.

Comparación con otros entornos

- **Google Sheets:** También tiene LAMBDA, LET, MAP, etc., con comportamientos similares a Excel 365. La diferencia principal es que Google Sheets tiende a ser más restrictivo en la recursión y el manejo de nombres, pero el concepto de "ciudadano de primera clase" se aplica de manera análoga.
- **Lenguajes funcionales (e.g., Haskell):** En estos, las funciones son plenamente ciudadanos de primera clase, con capacidades como composición y retorno directo.
- **Excel clásico:** Antes de las funciones dinámicas, las funciones no eran ciudadanos de primera clase en absoluto.

Conclusión

Las funciones en Excel, particularmente las LAMBDA en Excel 365, pueden considerarse **ciudadanos de primera clase de manera limitada o condicional**. Cumplen con la mayoría de los criterios (asignación, paso como argumento, creación dinámica), pero la incapacidad de devolverlas como funciones invocables y la dependencia de nombres predefinidos las limita en comparación con lenguajes de programación tradicionales. Esto las convierte en un híbrido: poderosas dentro del paradigma de las hojas de cálculo, pero no tan flexibles como en un entorno de programación pura.

Si planeas usarlas extensivamente, nombrar LAMBDA y combinarlas con MAP , BYROW , y LET te dará una experiencia cercana a la programación funcional. ¿Te gustaría un ejemplo más específico de cómo usar esto en tu caso? ¡Estoy aquí para ayudarte! 😊

(Nota: La fecha y hora actuales son 12:27 PM CEST, jueves 25 de septiembre de 2025, según el sistema.)