# Lista 10: Numpy

Demuestras un buen conocimiento de la biblioteca numpy, empleándola de manera efectiva para resolver problemas relacionados con manipulación de arrays y operaciones matemáticas. Sin embargo, se pueden optimizar ciertos fragmentos para hacerlos más legibles y eficientes, así como aprovechar más funciones nativas de la biblioteca.

* **Cosas positivas:**
  + Manejo adecuado de las operaciones con arrays multidimensionales.
  + Uso creativo de funciones de numpy como reshape, add y split.
  + Implementación funcional de problemas interactivos y juegos.
* **Posibles mejoras:**
  + Optimizar el uso de bucles para aprovechar funciones vectorizadas de numpy.
  + Agregar más comentarios para explicar cada paso en los ejercicios más complejos.
  + Simplificar la lógica en ejercicios que involucran iteraciones anidadas.

## Ejercicios Obligatorios

### **Ejercicio 1**

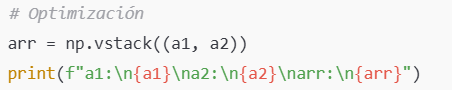
**Estado del ejercicio:** Correcto.  
**Errores encontrados:** Ninguno.  
**Código optimizado:**



**Comentario:** El ejercicio utiliza correctamente reshape. La optimización sugerida reduce líneas de código y mejora la legibilidad.

### **Ejercicio 2**

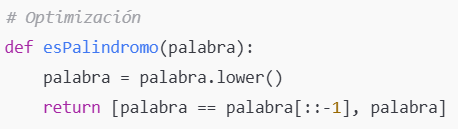
**Estado del ejercicio:** Correcto.  
**Errores encontrados:** Ninguno.  
**Código optimizado:**



**Comentario:** Buen uso de concatenate, aunque vstack simplifica el código y es más semántico para apilar arrays verticalmente.

### **Ejercicio 3**

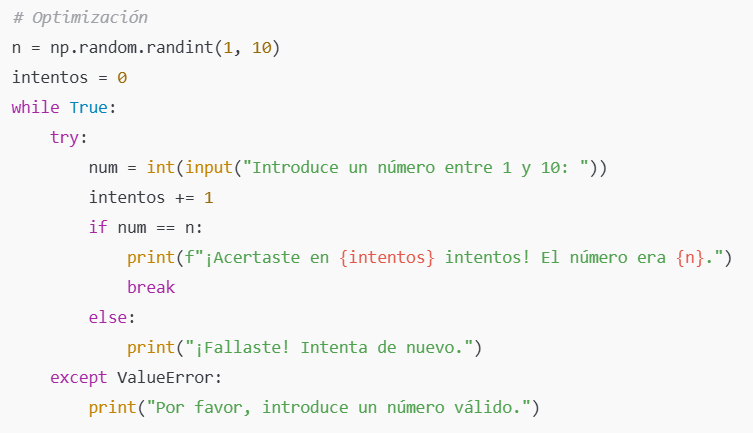
**Estado del ejercicio:** Incorrecto.  
**Errores encontrados:** El método str.lower(palabra) no modifica la cadena original, lo que causa errores en la lógica de comparación.  
**Código optimizado:**



**Comentario:** La lógica del palíndromo es funcional, pero debe usarse un enfoque más directo con slicing para garantizar precisión y simplicidad.

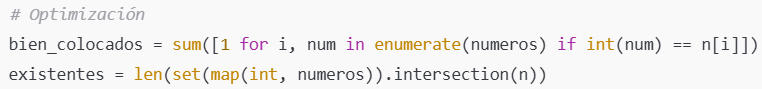
### **Ejercicio 4**

**Estado del ejercicio:** Correcto.  
**Errores encontrados:** Ninguno.  
**Código optimizado:**

**Comentario:** El ejercicio cumple con el objetivo, pero la optimización mejora la claridad y elimina bucles anidados.

### **Ejercicio 5**

**Estado del ejercicio:** Correcto.  
**Errores encontrados:** Ninguno.  
**Código optimizado:**

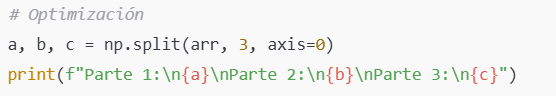
**Comentario:** Buen manejo de la lógica de comparación, pero usar conjuntos y comprensiones reduce la complejidad.

### **Ejercicio 6**

**Estado del ejercicio:** Correcto.  
**Errores encontrados:** Ninguno.  
**Código optimizado:** No es necesario.  
**Comentario:** Excelente implementación del uso de numpy.add para sumar matrices generadas aleatoriamente.

### **Ejercicio 7**

**Estado del ejercicio:** Correcto.  
**Errores encontrados:** Ninguno.  
**Código optimizado:**



**Comentario:** La división de matrices está bien implementada. La optimización mejora la presentación de los resultados.

## Ejercicios opcionales

### **Ejercicio 8**

Ejercicios opcionales no realizados