

# Introducción a la Programación

Julian Blanco, Lucas de Fino, Federico Giorgi, Dafydd Jenkins, Lara  
Rodrigues La Moglie, Natalia Massri, Federico Peitti

Primer semestre de 2025

Clase práctica 6: Listas

# Érase una vez en una galaxia muy lejana...

Son tiempos de guerra civil. Nuestras naves rebeldes han atacado desde una base secreta y han obtenido su primera victoria contra el malvado Imperio Galáctico. Durante la batalla, espías rebeldes lograron robar los planos secretos del arma más extrema del Imperio: la **Estrella de la Muerte**, una estación espacial blindada con suficiente potencia para destruir un planeta entero.

Por esto mismo nuestra nave **Halcón Milenario**, comandada por **Han Solo**, se encuentra en una misión especial: destruir la Estrella de la Muerte construida por el maléfico Lord Sith.

Cercanas a la Estrella se encuentran *naves estelares* protegiéndola, a la espera de la detección de intrusos rebeldes para destruirlos. Para evadir los radares de las naves estelares dispersas en su trayecto, el Halcón Milenario debe moverse lentamente y con sus escudos desactivados.

Nuestro objetivo es decidir si es necesario enviar una señal de alerta a Han Solo en su trayecto a la Estrella de la Muerte cuando se encuentre a una distancia crítica de las *naves estelares*.

# Érase una vez en una galaxia muy lejana...

Por suerte el **Halcón Milenario** cuenta con un sensor que detecta la proximidad que tiene con las *naves estelares*, este sensor nos informa rápidamente una **lista** de tamaño desconocido con todas las distancias que hay hasta una *nave estelar*.

Nuestra misión, como representantes de **Intro a la Programación** es que dada una lista **sensado** de enteros y un punto crítico **p** entero mayor o igual a cero, podamos decir si existe alguna *nave estelar* a una proximidad menor o igual al punto crítico.

Por ejemplo, sea la lista [32638, 205, 258, 71115] y el punto crítico 250, entonces nuestra función debe decir que sí, hay una *nave estelar* peligrosamente cerca.

1. Escribir la especificación (en `star_wars.py`) y luego los casos de test para este problema (en `tests_star_wars.py`).

# Érase una vez en una galaxia muy lejana...

Por suerte el **Halcón Milenario** cuenta con un sensor que detecta la proximidad que tiene con las *naves estelares*, este sensor nos informa rápidamente una **lista** de tamaño desconocido con todas las distancias que hay hasta una *nave estelar*.

Nuestra misión, como representantes de **Intro a la Programación** es que dada una lista **sensado** de enteros y un punto crítico **p** entero mayor o igual a cero, podamos decir si existe alguna *nave estelar* a una proximidad menor o igual al punto crítico.

Por ejemplo, sea la lista [32638, 205, 258, 71115] y el punto crítico 250, entonces nuestra función debe decir que sí, hay una *nave estelar* peligrosamente cerca.

1. Escribir la especificación (en `star_wars.py`) y luego los casos de test para este problema (en `tests_star_wars.py`).
2. Escribir un programa en Python que se adecúe a la especificación anteriormente construída (en `star_wars.py`).

# Érase una vez en una galaxia muy lejana...

Por suerte el **Halcón Milenario** cuenta con un sensor que detecta la proximidad que tiene con las *naves estelares*, este sensor nos informa rápidamente una **lista** de tamaño desconocido con todas las distancias que hay hasta una *nave estelar*.

Nuestra misión, como representantes de **Intro a la Programación** es que dada una lista **sensado** de enteros y un punto crítico **p** entero mayor o igual a cero, podamos decir si existe alguna *nave estelar* a una proximidad menor o igual al punto crítico.

Por ejemplo, sea la lista [32638, 205, 258, 71115] y el punto crítico 250, entonces nuestra función debe decir que sí, hay una *nave estelar* peligrosamente cerca.

1. Escribir la especificación (en `star_wars.py`) y luego los casos de test para este problema (en `tests_star_wars.py`).
2. Escribir un programa en Python que se adecúe a la especificación anteriormente construída (en `star_wars.py`).
3. Ejecutar el programa desde la consola.

# Érase una vez en una galaxia muy lejana...

Nuestra segunda y última misión (el resto queda en manos de **Han Solo**) es que dada una lista **sensado** y un punto crítico **p**, podamos brindar una lista con las distancias de las *naves estelares* que están a una proximidad menor o igual al punto crítico. La lista devuelta tiene que respetar la cantidad de ocurrencias de una distancia.

Por ejemplo, sea la lista [32638, 205, 258, 71115, 198, 205] y el punto crítico 250, entonces nuestra función debe devolver la lista [205, 198, 205].

# Érase una vez en una galaxia muy lejana...

Nuestra segunda y última misión (el resto queda en manos de **Han Solo**) es que dada una lista **sensado** y un punto crítico **p**, podamos brindar una lista con las distancias de las *naves estelares* que están a una proximidad menor o igual al punto crítico. La lista devuelta tiene que respetar la cantidad de ocurrencias de una distancia.

Por ejemplo, sea la lista [32638, 205, 258, 71115, 198, 205] y el punto crítico 250, entonces nuestra función debe devolver la lista [205, 198, 205].

1. Completar la especificación de la función **naves\_cercanas**.

# Érase una vez en una galaxia muy lejana...

Nuestra segunda y última misión (el resto queda en manos de **Han Solo**) es que dada una lista **sensado** y un punto crítico **p**, podamos brindar una lista con las distancias de las *naves estelares* que están a una proximidad menor o igual al punto crítico. La lista devuelta tiene que respetar la cantidad de ocurrencias de una distancia.

Por ejemplo, sea la lista [32638, 205, 258, 71115, 198, 205] y el punto crítico 250, entonces nuestra función debe devolver la lista [205, 198, 205].

1. Completar la especificación de la función **naves\_cercanas**.
2. Completar la implementación de la función de modo que respete la especificación previa.



# Érase una vez en una galaxia muy lejana...

Nuestra segunda y última misión (el resto queda en manos de **Han Solo**) es que dada una lista **sensado** y un punto crítico **p**, podamos brindar una lista con las distancias de las *naves estelares* que están a una proximidad menor o igual al punto crítico. La lista devuelta tiene que respetar la cantidad de ocurrencias de una distancia.

Por ejemplo, sea la lista [32638, 205, 258, 71115, 198, 205] y el punto crítico 250, entonces nuestra función debe devolver la lista [205, 198, 205].

1. Completar la especificación de la función **naves\_cercanas**.
2. Completar la implementación de la función de modo que respete la especificación previa.
3. Proponer un predicado invariante que describa el trabajo realizado por el ciclo involucrado.

# Que la fuerza de las listas te acompañe

Hoy vimos cómo:

- ▶ Utilizar las listas como otra herramienta a la hora de pensar problemas y escribir algoritmos.

# Que la fuerza de las listas te acompañe

Hoy vimos cómo:

- ▶ Utilizar las listas como otra herramienta a la hora de pensar problemas y escribir algoritmos.
- ▶ Pensar casos de test para verificar el algoritmo desarrollado.

# Que la fuerza de las listas te acompañe

Hoy vimos cómo:

- ▶ Utilizar las listas como otra herramienta a la hora de pensar problemas y escribir algoritmos.
- ▶ Pensar casos de test para verificar el algoritmo desarrollado.
- ▶ Ejecutar programas desde la consola.

# Que la fuerza de las listas te acompañe

Hoy vimos cómo:

- ▶ Utilizar las listas como otra herramienta a la hora de pensar problemas y escribir algoritmos.
- ▶ Pensar casos de test para verificar el algoritmo desarrollado.
- ▶ Ejecutar programas desde la consola.

Con lo visto, ya pueden resolver hasta el ejercicio 4 (inclusive) de la Guía 5.