

# Introducción a coherencia computacional, segundo semestre del 2022, información del curso

Luis Ross Lépez

3 de agosto de 2022

Versión 1.0



# Índice

|   |          |
|---|----------|
| <b>1. Descripción general</b>   | <b>3</b> |
| <b>2. Objetivos</b>   | <b>3</b> |
| 2.1. Objetivo General . . . . .   | 3        |
| 2.2. Objetivos Específicos . . . . .  | 3        |
| <b>3. Contenidos</b>  | <b>4</b> |
| 3.1. Conocimiento general sobre qué es cohetería computacional . . . . .                    | 4        |
| 3.2. Fundamentos tras los lenguajes de programación y sus diferencias . . . . .             | 4        |
| 3.3. Taller básico de programación en Python con enfoque a aprender más lenguajes . . . . . | 4        |
| <b>4. Nota final del curso</b>  | <b>5</b> |

## 1. Descripción general

Introducción a cohetería computacional tiene como intención brindarle a los estudiantes una idea sobre el conocimiento y trabajo que va detrás de diseñar un sistema computacional para una misión de cohetería, esto desde los aspectos de electrónica, programación y un poco de organización de trabajo en equipo.

El estudiante que disfrute de esta área y aproveche el curso tendrá herramientas útiles para todo momento en programación así como conocimiento general sobre cómo funcionan todos los equipos que nos rodean día a día desde el celular en nuestras manos, hasta los chips que envían información desde satélites y aparatos espaciales en el espacio exterior.

## 2. Objetivos

### 2.1. Objetivo General

Brindar el conocimiento necesario de programación, electrónica y telemetría para llevar a cabo una misión de cohetería exitosa, donde se diseñe una computadora para el cohete, una aplicación para control de misión y un sistema de recuperación que facilite la misma, así aportando a la misión en toda etapa.

### 2.2. Objetivos Específicos

1. Enseñar los fundamentos detrás de los lenguajes de programación que se usan en la actualidad, así como un trasfondo histórico sobre los mismos.
2. Aprender a programar en lenguajes de alto nivel bastante comunes como lo son Python y Arduino, para poder programar tanto en computadoras comunes como en microcontroladores.
3. Diseñar circuitos básicos con conceptos simples de corriente continua y el uso de microcontroladores programables para automatizar tareas.
4. Usar sensores y actuadores según sea necesario para las soluciones de electrónica que se consideren necesarias en los circuitos creados.
5. Conocer la teoría de telemetría y poder aplicarla para realizar conexiones inalámbricas y transmisión de datos a distancia en los dispositivos elegidos.
6. Programar una aplicación pensada para el uso de un usuario promedio, con funcionalidades útiles o llamativas antes, durante y después del lanzamiento y aterrizaje del cohete.
7. Trabajar en conjunto para diseñar un sistema completo e inter funcional de electrónica y programación.
8. Ganar conocimiento de programación que pueda ser aplicado a cualquier lenguaje que se encuentren en un futuro en su carrera profesional.

### **3. Contenidos**

#### **3.1. Conocimiento general sobre qué es cohetería computacional**

- Hardware a lo largo de la historia.
- Software a lo largo de la historia.
- Computación en el espacio (NASA y SpaceX).
- Objetivos de cohetería computacional en TECSpace.
- Tecnologías que usamos en TECSpace.
- Ejemplos de proyectos existentes y viables.

#### **3.2. Fundamentos tras los lenguajes de programación y sus diferencias**

- Diferencias entre lenguajes de alto y bajo nivel.
- Paradigmas de programación y sus usos.
- Diferencias entre lenguajes compilados e interpretados.
- Compatibilidad de sistemas operativos.
- Compatibilidad de navegadores web.
- Compatibilidad de dispositivos (arquitecturas).
- Pseudocódigo y diagramas de flujo.

#### **3.3. Taller básico de programación en Python con enfoque a aprender más lenguajes**

- Qué son variables detalles y cómo usarlas.
- Instanciación de variables vs declaración de valores.
- Utilidad de funciones, cómo usarlas y sus partes.
- Código estructurado para manejo de pruebas lógicas.
- Ciclos de código do while, y do for.
- Listas en Python sus usos y cómo se diferencian de arrays.
- Cómo importar código de otros archivos en Python (módulos).
- Lectura y escritura de datos en archivos de texto I/O.
- Cómo facilitar el aprendizaje u entendimiento de lenguajes que no conocemos.

## 4. Nota final del curso

Como se indicó previamente, la nota final del curso para cada estudiante constará tanto de la nota grupal del proyecto, como de otros valores, estos se detallan a continuación.

| Item a evaluar         | Valor del item [%] |
|------------------------|--------------------|
| Proyecto computacional | 55                 |
| Coevaluación proyecto  | 10                 |
| Tareas computacional   | 35                 |

Todo estudiante que obtenga una nota de 70 será considerado aprobado del curso de cohetería computacional, esto significa que podrá participar en los proyectos de TECSpace como integrante y recibirá un reconocimiento por haber aprobado el curso, el cual puede ser usado para demostrar experiencia y dedicación.