

## TP #2 Herramientas para Desarrollo

### Simulación

1. ¿Cómo se utiliza Proteus para simular circuitos electrónicos que incluyen microcontroladores?
2. ¿Cuáles son las ventajas de utilizar la simulación electrónica en el diseño y desarrollo de circuitos electrónicos?
3. ¿Qué técnicas y herramientas se utilizan en la simulación de circuitos electrónicos digitales?
4. ¿Cómo se modelan y simulan los circuitos eléctricos en Proteus?
5. ¿Cuáles son las limitaciones de la simulación electrónica y cómo se pueden mitigar?

### Programación

1. ¿Cuál es la diferencia entre una IDE y un editor de texto?
2. ¿Cuál es la ventaja de usar una IDE sobre un editor de texto al programar en Python?
3. ¿Qué es un entorno virtual en Python y para qué se utiliza?
4. ¿Cómo se declara una variable en C y cuál es la diferencia entre una variable global y una local?
5. ¿Cuál es la función de los archivos de cabecera en C?
6. ¿Cómo se compila y se ejecuta un programa en C utilizando Visual Studio Code?
7. ¿Qué es Google Colab y cómo se utiliza para programar en Python?
8. ¿Qué son las librerías o bibliotecas en programación y cuál es su importancia en el desarrollo de software?
9. Instale VScode y configúrelo con las extensiones adecuadas
10. Cree una cuenta en Google, cree un cuaderno con Colab

### Metodologías

1. ¿Qué es el manifiesto ágil?
2. ¿Cuáles son los valores fundamentales del manifiesto ágil?
3. ¿Qué es Scrum y cuáles son sus roles principales?
4. ¿Qué es Kanban y cómo se utiliza en metodologías ágiles?
5. ¿Qué es el sprint en Scrum?
6. ¿Qué es un tablero Kanban y cuál es su función?
7. ¿Cuáles son los principios del desarrollo ágil de software?
8. ¿Qué es la retrospectiva en Scrum y para qué se utiliza?
9. ¿Qué es el flujo continuo y cómo se relaciona con Kanban?
10. ¿Qué es Git y cómo se utiliza en metodologías ágiles?

11. ¿Qué es GitHub y para qué se utiliza en el desarrollo de software?
12. ¿Cuáles son algunas de las características clave de GitHub?
13. ¿Cómo se utiliza GitHub para colaborar en proyectos de código abierto?
14. ¿Cuál es la diferencia entre un repositorio público y privado en GitHub?
15. ¿Cómo se utilizan las ramas (branches) en GitHub y por qué son útiles?
16. ¿Qué es Git y cuál es su función principal?
17. ¿Cuál es la diferencia entre "commit" y "push" en Git?
18. ¿Qué es un "branch" en Git y para qué se utiliza?
19. ¿Cómo se fusionan dos ramas en Git?
20. ¿Cómo se solucionan conflictos de merge en Git?