**ACTIVIDAD PRÁCTICA # 1**

**Actividades:**

**Actividad 1: Variación del Contrato de Almacenamiento:**

* **Descripción:** Modifica el contrato SimpleStorage para que además de almacenar un valor entero, también permita a los usuarios recuperar el valor almacenado.
	+ **Comandos Principales:**

setData(uint256 \_data): Función para establecer el valor almacenado.

data(): Función para recuperar el valor almacenado.

* + **Explicación:**

setData: Permite a los usuarios establecer el valor almacenado en el contrato.

data: Permite a los usuarios recuperar el valor almacenado del contrato.

**Actividad 2: Variación del Contrato de Transferencia de Ether:**

* **Descripción:** Modifica el contrato SimpleTransfer para que además de permitir a los usuarios transferir Ether a otra dirección, también tenga una función que devuelva el saldo actual del contrato.
	+ **Comandos Principales:**

transfer(address payable \_recipient) payable: Función para transferir Ether a otra dirección.

getContractBalance(): Función para consultar el saldo actual del contrato.

* + **Explicación:**

transfer: Permite a los usuarios enviar Ether a otra dirección especificada.

getContractBalance: Permite a los usuarios consultar el saldo actual del contrato.

**Actividad 3: Variación del Contrato de Verificación de Paridad:**

* **Descripción:** Modifica el contrato ParityVerifier para que tenga una función que verifique si un número dado es primo.
	+ **Comandos Principales:**

checkPrime(uint256 \_number) pure returns (bool): Función para verificar si un número dado es primo.

* + **Explicación:**

checkPrime: Verifica si el número dado es primo y devuelve true si lo es, y false en caso contrario.