

Desarrollo Blockchain Ethereum con Solidity

Módulo 3 - Manejo de errores

Manejo de errores

Excepciones

- Las excepciones ocurren en sub-llamadas y deben ser gestionadas por medio de la sentencia **try-catch**.
- Hay que tener en cuenta que funciones como **call** y **delegate call** no lanzan excepción si la dirección de destino no existe.
- Para chequeos que pueden prevenir errores se pueden utilizar **assert** y **require**.
- **Assert** se utiliza para pruebas internas.
- **Require** es recomendado ya que revierte la transacción y lanza un Error.
- **Error(string)** es el que nos muestra un mensaje descriptivo de la situación inesperada.
- También se puede llamar al **revert(mensaje)** directamente.



Excepciones

```
interface DataFeed { function getData(address token) external returns (uint value); }

contract FeedConsumer {
    DataFeed feed;
    uint errorCount;
    function rate(address token) public returns (uint value, bool success) {
        // Permanently disable the mechanism if there are
        // more than 10 errors.
        require(errorCount < 10);
        try feed.getData(token) returns (uint v) {
            return (v, true);
        } catch Error(string memory /*reason*/) {
            // This is executed in case
            // revert was called inside getData
            // and a reason string was provided.
            errorCount++;
            return (0, false);
        } catch Panic(uint /*errorCode*/) {
            // This is executed in case of a panic,
            // i.e. a serious error like division by zero
            // or overflow. The error code can be used
            // to determine the kind of error.
            errorCount++;
            return (0, false);
        } catch (bytes memory /*lowLevelData*/) {
            // This is executed in case revert() was used.
            errorCount++;
            return (0, false);
        }
    }
}
```

Logs en Solidity

- Para tener un log efectivo en Solidity se debe implementar por medio de Eventos.
- Estos logs no serán accesibles desde el contrato y deben ser accedidos desde algún medio externo.
- Recordar que la emisión de eventos infiere un gasto de gas, por lo cual hay que tener criterio a la hora de emitirlos.

```
eventExample.once("NombreDelEvento", (error, event) => {  
    if (!error) console.log(event);  
});
```



¡Muchas gracias!

¡Sigamos trabajando!