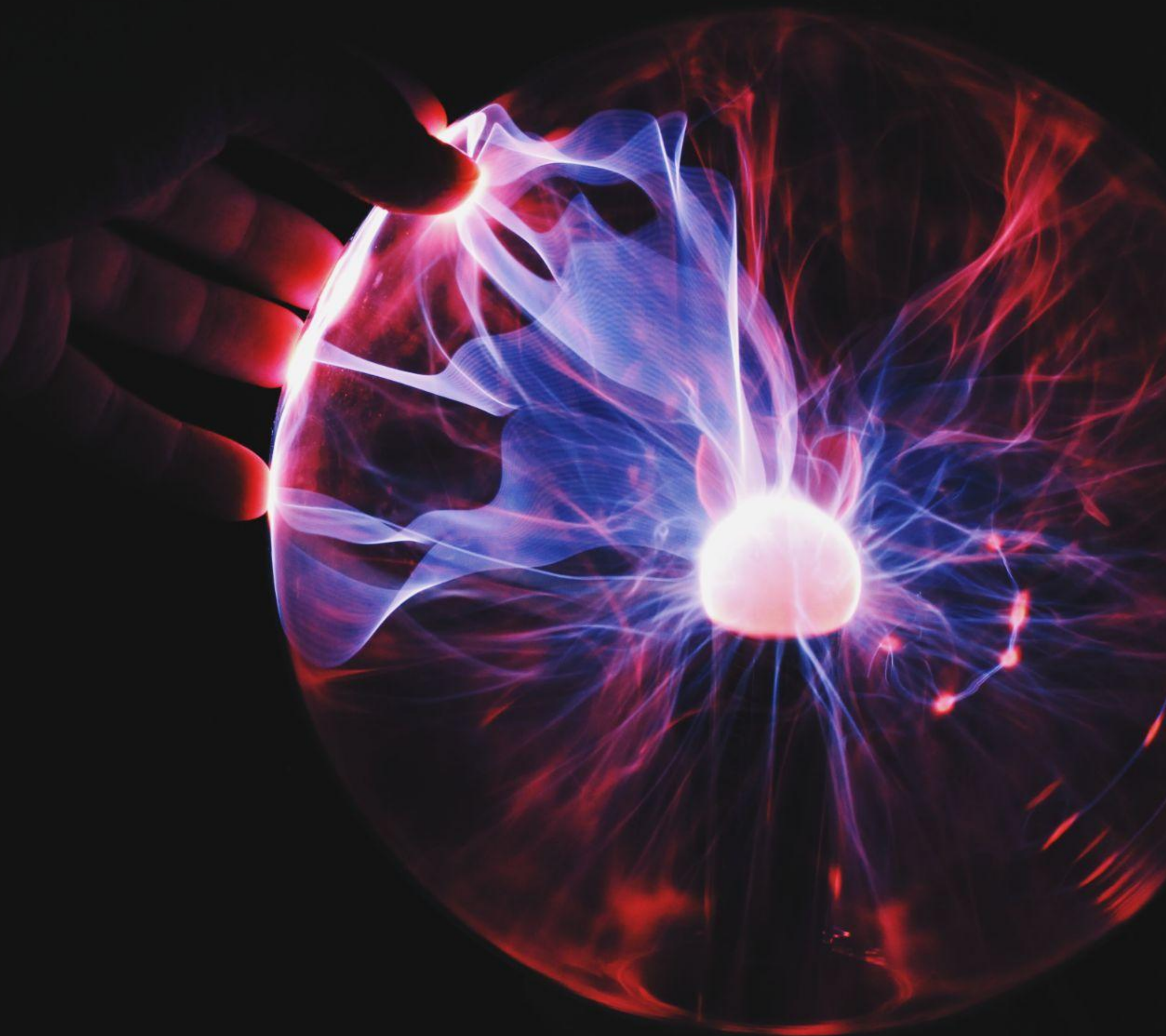
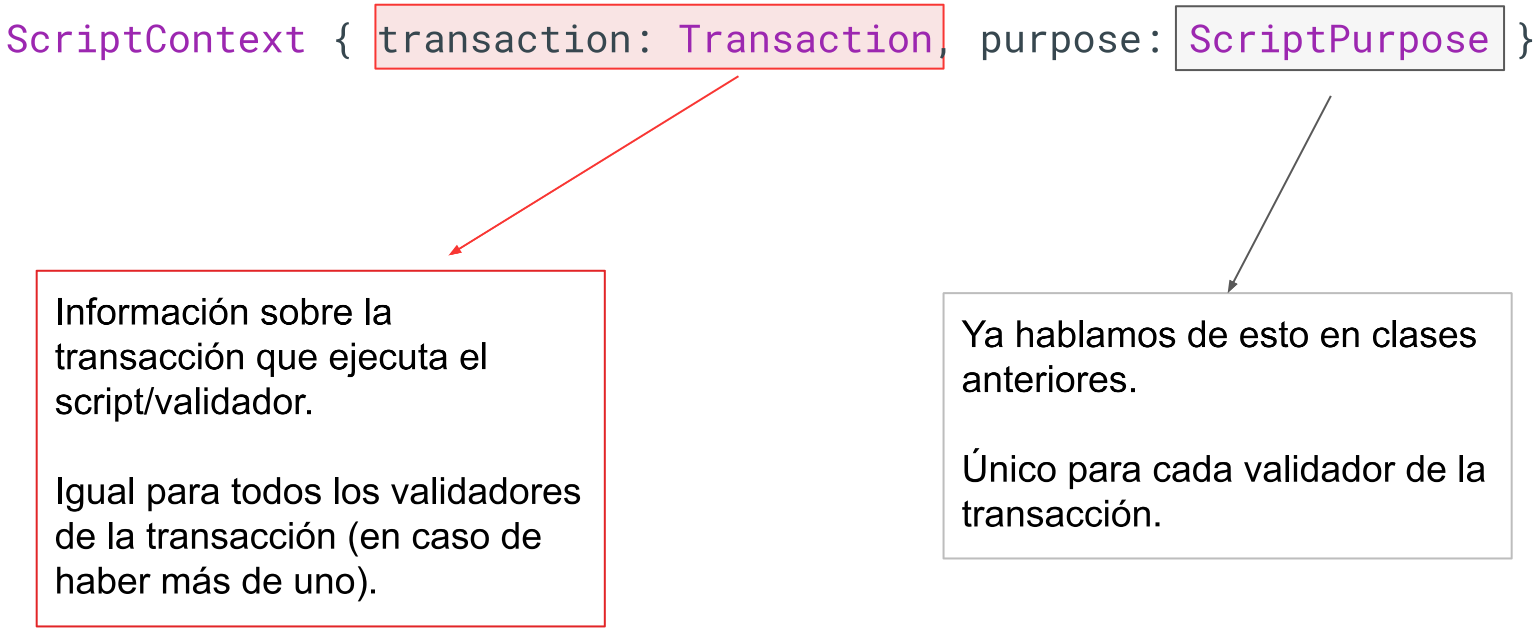
INPUT | OUTPUT



Contexto de la Transacción

Script Context: Qué contiene el contexto del script?



```
ScriptContext { transaction: Transaction, purpose: ScriptPurpose }
```

Información sobre la transacción que ejecuta el script/validador.

Igual para todos los validadores de la transacción (en caso de haber más de uno).

Ya hablamos de esto en clases anteriores.

Único para cada validador de la transacción.

Contexto de la Tx: Qué contiene `Transaction`?

```
Transaction {
  inputs: List<Input>,
  reference_inputs: List<Input>,
  outputs: List<Output>,
  fee: Value,
  mint: MintedValue,
  certificates: List<Certificate>,
  withdrawals: Pairs<StakeCredential, Int>,
  validity_range: ValidityRange,
  extra_signatories: List<Hash<Blake2b_224, VerificationKey>>,
  redeemers: Pairs<ScriptPurpose, Redeemer>,
  datums: Dict<Hash<Blake2b_256, Data>, Data>,
  id: TransactionId,
}
```

Contexto de la Tx: Qué contiene `Transaction`?

```
Transaction {
  inputs: List<Input>,
  reference_inputs: List<Input>,
  outputs: List<Output>,
  fee: Value,
  mint: MintedValue,
  certificates: List<Certificate>,
  withdrawals: Pairs<StakeCredential, Int>,
  validity_range: ValidityRange,
  extra_signatories: List<Hash<Blake2b_224, VerificationKey>>,
  redeemers: Pairs<ScriptPurpose, Redeemer>,
  datums: Dict<Hash<Blake2b_256, Data>, Data>,
  id: TransactionId,
}
```

Lista de todos los UTxO que consume la transacción

Contexto de la Tx: Qué contiene `Transaction`?

```
Transaction {
  inputs: List<Input>,
  reference_inputs: List<Input>,
  outputs: List<Output>,
  fee: Value,
  mint: MintedValue,
  certificates: List<Certificate>,
  withdrawals: Pairs<StakeCredential, Int>,
  validity_range: ValidityRange,
  extra_signatories: List<Hash<Blake2b_224, VerificationKey>>,
  redeemers: Pairs<ScriptPurpose, Redeemer>,
  datums: Dict<Hash<Blake2b_256, Data>, Data>,
  id: TransactionId,
}
```

Lista de todos los UTxO que la Tx no consume pero puede ver

Contexto de la Tx: Qué contiene Transaction?

```
Transaction {
  inputs: List<Input>,
  reference_inputs: List<Input>,
  outputs: List<Output>,
  fee: Value,
  mint: MintedValue,
  certificates: List<Certificate>,
  withdrawals: Pairs<StakeCredential, Int>,
  validity_range: ValidityRange,
  extra_signatories: List<Hash<Blake2b_224, VerificationKey>>,
  redeemers: Pairs<ScriptPurpose, Redeemer>,
  datums: Dict<Hash<Blake2b_256, Data>, Data>,
  id: TransactionId,
}
```

Lista de los UTxO que la transacción crea

Contexto de la Tx: Qué contiene Transaction?

```
Transaction {
  inputs: List<Input>,
  reference_inputs: List<Input>,
  outputs: List<Output>,
  fee: Value,
  mint: MintedValue,
  certificates: List<Certificate>,
  withdrawals: Pairs<StakeCredential, Int>,
  validity_range: ValidityRange,
  extra_signatories: List<Hash<Blake2b_224, VerificationKey>>,
  redeemers: Pairs<ScriptPurpose, Redeemer>,
  datums: Dict<Hash<Blake2b_256, Data>, Data>,
  id: TransactionId,
}
```

Lo que hay que pagar al nodo por validar y procesar la Tx

Contexto de la Tx: Qué contiene `Transaction`?

```
Transaction {
  inputs: List<Input>,
  reference_inputs: List<Input>,
  outputs: List<Output>,
  fee: Value,
  mint: MintedValue,
  certificates: List<Certificate>,
  withdrawals: Pairs<StakeCredential, Int>,
  validity_range: ValidityRange,
  extra_signatories: List<Hash<Blake2b_224, VerificationKey>>,
  redeemers: Pairs<ScriptPurpose, Redeemer>,
  datums: Dict<Hash<Blake2b_256, Data>, Data>,
  id: TransactionId,
}
```

Tokens minteados y/o quemados en esta Tx

Contexto de la Tx: Qué contiene `Transaction`?

```
Transaction {
  inputs: List<Input>,
  reference_inputs: List<Input>,
  outputs: List<Output>,
  fee: Value,
  mint: MintedValue,
  certificates: List<Certificate>,
  withdrawals: Pairs<StakeCredential, Int>,
  validity_range: ValidityRange,
  extra_signatories: List<Hash<Blake2b_224, VerificationKey>>,
  redeemers: Pairs<ScriptPurpose, Redeemer>,
  datums: Dict<Hash<Blake2b_256, Data>, Data>,
  id: TransactionId,
}
```

Lista de certificados atestando una operación (registrar una pool, delegar, gobernanza…)

Contexto de la Tx: Qué contiene Transaction?

```
Transaction {
  inputs: List<Input>,
  reference_inputs: List<Input>,
  outputs: List<Output>,
  fee: Value,
  mint: MintedValue,
  certificates: List<Certificate>,
  withdrawals: Pairs<StakeCredential, Int>,
  validity_range: ValidityRange,
  extra_signatories: List<Hash<Blake2b_224, VerificationKey>>,
  redeemers: Pairs<ScriptPurpose, Redeemer>,
  datums: Dict<Hash<Blake2b_256, Data>, Data>,
  id: TransactionId,
}
```

Lista de pares que contienen credenciales de staking con los montos de Lovelace que están retirando en esta transacción

Contexto de la Tx: Qué contiene Transaction?

```
Transaction {
  inputs: List<Input>,
  reference_inputs: List<Input>,
  outputs: List<Output>,
  fee: Value,
  mint: MintedValue,
  certificates: List<Certificate>,
  withdrawals: Pairs<StakeCredential, Int>,
  validity_range: ValidityRange,
  extra_signatories: List<Hash<Blake2b_224, VerificationKey>>,
  redeemers: Pairs<ScriptPurpose, Redeemer>,
  datums: Dict<Hash<Blake2b_256, Data>, Data>,
  id: TransactionId,
}
```

Rango de tiempo durante el cual la Tx es válida

Contexto de la Tx: Qué contiene `Transaction`?

```
Transaction {
  inputs: List<Input>,
  reference_inputs: List<Input>,
  outputs: List<Output>,
  fee: Value,
  mint: MintedValue,
  certificates: List<Certificate>,
  withdrawals: Pairs<StakeCredential, Int>,
  validity_range: ValidityRange,
  extra_signatories: List<Hash<Blake2b_224, VerificationKey>>,
  redeemers: Pairs<ScriptPurpose, Redeemer>,
  datums: Dict<Hash<Blake2b_256, Data>, Data>,
  id: TransactionId,
}
```

Lista de hashes
públicos que firmaron
la transacción

Contexto de la Tx: Qué contiene `Transaction`?

```
Transaction {
  inputs: List<Input>,
  reference_inputs: List<Input>,
  outputs: List<Output>,
  fee: Value,
  mint: MintedValue,
  certificates: List<Certificate>,
  withdrawals: Pairs<StakeCredential, Int>,
  validity_range: ValidityRange,
  extra_signatories: List<Hash<Blake2b_224, VerificationKey>>,
  redeemers: Pairs<ScriptPurpose, Redeemer>,
  datums: Dict<Hash<Blake2b_256, Data>, Data>,
  id: TransactionId,
}
```

Lista de pares que contienen los propósitos de todos los scripts de la Tx con sus respectivos redeemers

Contexto de la Tx: Qué contiene `Transaction`?

```
Transaction {
  inputs: List<Input>,
  reference_inputs: List<Input>,
  outputs: List<Output>,
  fee: Value,
  mint: MintedValue,
  certificates: List<Certificate>,
  withdrawals: Pairs<StakeCredential, Int>,
  validity_range: ValidityRange,
  extra_signatories: List<Hash<Blake2b_224, VerificationKey>>,
  redeemers: Pairs<ScriptPurpose, Redeemer>,
  datums: Dict<Hash<Blake2b_256, Data>, Data>,
  id: TransactionId,
}
```

Diccionario que relaciona los hashes de datums con los datums adjuntados a la transacción

Contexto de la Tx: Qué contiene `Transaction`?

```
Transaction {
  inputs: List<Input>,
  reference_inputs: List<Input>,
  outputs: List<Output>,
  fee: Value,
  mint: MintedValue,
  certificates: List<Certificate>,
  withdrawals: Pairs<StakeCredential, Int>,
  validity_range: ValidityRange,
  extra_signatories: List<Hash<Blake2b_224, VerificationKey>>,
  redeemers: Pairs<ScriptPurpose, Redeemer>,
  datums: Dict<Hash<Blake2b_256, Data>, Data>,
  id: TransactionId,
}
```

Identificador único de la transacción. Hash de la transacción serializada

Contexto de la Tx: Qué contiene `Transaction`?

```
Transaction {
  inputs: List<Input>,
  reference_inputs: List<Input>,
  outputs: List<Output>,
  fee: Value,
  mint: MintedValue,
  certificates: List<Certificate>,
  withdrawals: Pairs<StakeCredential, Int>,
  validity_range: ValidityRange,
  extra_signatories: List<Hash<Blake2b_224, VerificationKey>>,
  redeemers: Pairs<ScriptPurpose, Redeemer>,
  datums: Dict<Hash<Blake2b_256, Data>, Data>,
  id: TransactionId,
}
```

Fin

Preguntas?

INPUT | OUTPUT