



# Índice de Contenidos

## 1 Introducción

- Blockchains, Criptomonedas y Smart contracts
- Cardano
- ACTUS
- Verificación formal

## 2 Escribiendo contratos ACTUS en Cardano

- Notación del estándar ACTUS
- Contratos en Cardano

### 3 Verificando propiedades en contratos en Marlowe

- El modelo de Marlowe
- Pruebas sencillas sobre contratos específicos
- Warnings en Auction

## 4 Conclusión

- Resumen
- Posibles temas de desarrollo futuro

## 5 Bibliografía

# Índice de contenidos

- 1 **Introducción**
  - Blockchains, Criptomonedas y Smart contracts
    - Cardano
    - ACTUS
    - Verificación formal
- 2 **Escribiendo contratos ACTUS en Cardano**
  - Notación del estándar ACTUS
  - Contratos en Cardano
- 3 **Verificando propiedades en contratos en Marlowe**
  - El modelo de Marlowe
  - Pruebas sencillas sobre contratos específicos
  - Warnings en Auction
- 4 **Conclusión**
  - Resumen
  - Posibles temas de desarrollo futuro
- 5 **Bibliografía**

# Cadenas de bloques o *Blockchains*

Las cadenas de bloques, conocidas en inglés como blockchains, son estructuras de datos en las cuales la información se divide en conjuntos (bloques) que cuentan con información adicional relativa a bloques previos de la cadena.

# Cadenas de bloques o *Blockchains*

Las cadenas de bloques, conocidas en inglés como blockchains, son estructuras de datos en las cuales la información se divide en conjuntos (bloques) que cuentan con información adicional relativa a bloques previos de la cadena.

Con esta organización relativa, y con ayuda de técnicas criptográficas, la información de un bloque solo puede ser alterada modificando todos los bloques anteriores.

# Cadenas de bloques o *Blockchains*

Las cadenas de bloques, conocidas en inglés como blockchains, son estructuras de datos en las cuales la información se divide en conjuntos (bloques) que cuentan con información adicional relativa a bloques previos de la cadena.

Con esta organización relativa, y con ayuda de técnicas criptográficas, la información de un bloque solo puede ser alterada modificando todos los bloques anteriores.

Esta propiedad facilita su aplicación en un entorno distribuido, de manera tal que la cadena de bloques puede modelar una base de datos pública no relacional, que contenga un registro histórico irrefutable de información.

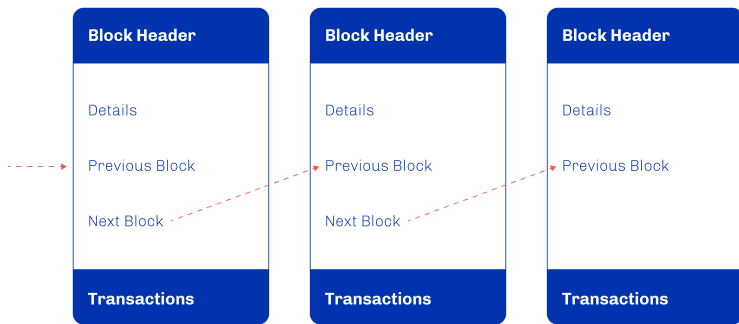
# Cadenas de bloques o *Blockchains*

Las cadenas de bloques, conocidas en inglés como blockchains, son estructuras de datos en las cuales la información se divide en conjuntos (bloques) que cuentan con información adicional relativa a bloques previos de la cadena.

Con esta organización relativa, y con ayuda de técnicas criptográficas, la información de un bloque solo puede ser alterada modificando todos los bloques anteriores.

Esta propiedad facilita su aplicación en un entorno distribuido, de manera tal que la cadena de bloques puede modelar una base de datos pública no relacional, que contenga un registro histórico irrefutable de información.

En la práctica esta técnica ha permitido la implementación de un registro contable o ledger distribuido que soporta y garantiza la seguridad de transacciones y dinero digital.



**Figura:** Representación simplificada de los datos en un bloque de la cadena. Extraída de [Brünjes and Vinogradova, 2019].



# Criptomonedas

Las criptomonedas son activos digitales que se almacenan en el ledger y están diseñadas para servir como medio de intercambio de bienes o servicios.

# Criptomonedas

Las criptomonedas son activos digitales que se almacenan en el ledger y están diseñadas para servir como medio de intercambio de bienes o servicios.

Los ledgers de blockchain son utilizados como tecnología subyacente para la creación de criptomonedas en un entorno descentralizado.

# Criptomonedas

Las criptomonedas son activos digitales que se almacenan en el ledger y están diseñadas para servir como medio de intercambio de bienes o servicios.

Los ledgers de blockchain son utilizados como tecnología subyacente para la creación de criptomonedas en un entorno descentralizado.

Los protocolos de blockchain utilizan técnicas criptográficas rigurosas para permitir la creación de criptomonedas, asegurar y verificar la propiedad de las mismas y los registros de movimiento de fondos.

# Criptomonedas

Las criptomonedas son activos digitales que se almacenan en el ledger y están diseñadas para servir como medio de intercambio de bienes o servicios.

Los ledgers de blockchain son utilizados como tecnología subyacente para la creación de criptomonedas en un entorno descentralizado.

Los protocolos de blockchain utilizan técnicas criptográficas rigurosas para permitir la creación de criptomonedas, asegurar y verificar la propiedad de las mismas y los registros de movimiento de fondos.

El precio de la criptomoneda no está controlado por un gobierno o una institución financiera centralizada. Se define por su valor, la correlación con las cifras del mundo real y está impulsado por la oferta y la demanda del mercado.

# Smart contracts

Un contrato inteligente o *smart contract* es un acuerdo digital automatizado.

# Smart contracts

Un contrato inteligente o *smart contract* es un acuerdo digital automatizado.

Los mismos están escritos en código, rastrean, verifican y ejecutan las transacciones de un contrato entre varias partes.

# Smart contracts

Un contrato inteligente o *smart contract* es un acuerdo digital automatizado.

Los mismos están escritos en código, rastrean, verifican y ejecutan las transacciones de un contrato entre varias partes.

Las transacciones del contrato se ejecutan automáticamente mediante el código del smart contract cuando se cumplen las condiciones predeterminadas. Esencialmente, un contrato inteligente es un programa cuyas entradas y salidas son acciones en una cadena de bloques.

# Smart contracts

Un contrato inteligente o *smart contract* es un acuerdo digital automatizado.

Los mismos están escritos en código, rastrean, verifican y ejecutan las transacciones de un contrato entre varias partes.

Las transacciones del contrato se ejecutan automáticamente mediante el código del smart contract cuando se cumplen las condiciones predeterminadas. Esencialmente, un contrato inteligente es un programa cuyas entradas y salidas son acciones en una cadena de bloques.

Los smart contracts son auto-ejecutables y no requieren las acciones o la presencia de terceros. El código del contrato inteligente se almacena y distribuye en la *blockchain*, lo que lo hace transparente e irreversible.



# Índice de contenidos

## 1 Introducción

- Blockchains, Criptomonedas y Smart contracts
- **Cardano**
- ACTUS
- Verificación formal

## 2 Escribiendo contratos ACTUS en Cardano

- Notación del estándar ACTUS
- Contratos en Cardano

## 3 Verificando propiedades en contratos en Marlowe

- El modelo de Marlowe
- Pruebas sencillas sobre contratos específicos
- Warnings en Auction

## 4 Conclusión

- Resumen
- Posibles temas de desarrollo futuro

## 5 Bibliografía

Cardano es una plataforma blockchain descentralizada de tercera generación, cuya criptomoneda es llamada *ada*.

Cardano es una plataforma blockchain descentralizada de tercera generación, cuya criptomoneda es llamada *ada*.

- **La primera generación** de blockchains (con Bitcoin como gran representante) ofrece ledgers descentralizados para la transferencia segura de criptomonedas.

Cardano es una plataforma blockchain descentralizada de tercera generación, cuya criptomoneda es llamada *ada*.

- **La primera generación** de blockchains (con Bitcoin como gran representante) ofrece ledgers descentralizados para la transferencia segura de criptomonedas. Sin embargo, tales *blockchains* no proporcionaron un entorno funcional para la liquidación de acuerdos complejos y el desarrollo de aplicaciones descentralizadas (dApps).

Cardano es una plataforma blockchain descentralizada de tercera generación, cuya criptomoneda es llamada *ada*.

- **La primera generación** de blockchains (con Bitcoin como gran representante) ofrece ledgers descentralizados para la transferencia segura de criptomonedas. Sin embargo, tales *blockchains* no proporcionaron un entorno funcional para la liquidación de acuerdos complejos y el desarrollo de aplicaciones descentralizadas (dApps).
- **La segunda generación** (cuyo ejemplo más conocido es Ethereum) proporcionó soluciones mejoradas para redactar y ejecutar contratos inteligentes, desarrollar aplicaciones y crear diferentes tipos de tokens.

Cardano es una plataforma blockchain descentralizada de tercera generación, cuya criptomoneda es llamada *ada*.

- **La primera generación** de blockchains (con Bitcoin como gran representante) ofrece ledgers descentralizados para la transferencia segura de criptomonedas. Sin embargo, tales *blockchains* no proporcionaron un entorno funcional para la liquidación de acuerdos complejos y el desarrollo de aplicaciones descentralizadas (dApps).
- **La segunda generación** (cuyo ejemplo más conocido es Ethereum) proporcionó soluciones mejoradas para redactar y ejecutar contratos inteligentes, desarrollar aplicaciones y crear diferentes tipos de tokens. Sin embargo, la segunda generación de cadenas de bloques a menudo enfrenta problemas en términos de escalabilidad.

# Cardano como *blockchain* de 3ra generación

Cardano se concibe como la cadena de bloques de tercera generación.

La misma combina las propiedades de las generaciones anteriores y ofrece las siguientes propiedades:

# Cardano como *blockchain* de 3ra generación

Cardano se concibe como la cadena de bloques de tercera generación.

La misma combina las propiedades de las generaciones anteriores y ofrece las siguientes propiedades:

- **Seguridad**



# Cardano como *blockchain* de 3ra generación

Cardano se concibe como la cadena de bloques de tercera generación.

La misma combina las propiedades de las generaciones anteriores y ofrece las siguientes propiedades:

- **Seguridad**
- **Escalabilidad:** Rendimiento de transacciones, escala de datos, ancho de banda de la red.

# Cardano como *blockchain* de 3ra generación

Cardano se concibe como la cadena de bloques de tercera generación.

La misma combina las propiedades de las generaciones anteriores y ofrece las siguientes propiedades:

- **Seguridad**
- **Escalabilidad:** Rendimiento de transacciones, escala de datos, ancho de banda de la red.
- **Funcionalidad:** Además del procesamiento de transacciones, la cadena de bloques debe proporcionar todos los medios para la liquidación de acuerdos comerciales.

# Cardano como *blockchain* de 3ra generación

Cardano se concibe como la cadena de bloques de tercera generación.

La misma combina las propiedades de las generaciones anteriores y ofrece las siguientes propiedades:

- **Seguridad**
- **Escalabilidad:** Rendimiento de transacciones, escala de datos, ancho de banda de la red.
- **Funcionalidad:** Además del procesamiento de transacciones, la cadena de bloques debe proporcionar todos los medios para la liquidación de acuerdos comerciales.
- **Desarrollo e Integración:** Es importante asegurarse que la blockchain esté en constante desarrollo en términos de sostenibilidad y sea interoperable con otras blockchains e instituciones financieras.

# Ada como criptomoneda de Cardano

Cada ledger de blockchain tiene su criptomoneda subyacente o moneda nativa. Ada es la moneda nativa o principal en Cardano. Esto significa que ada es la principal unidad de pago en Cardano.

# Ada como criptomoneda de Cardano

Cada ledger de blockchain tiene su criptomoneda subyacente o moneda nativa. Ada es la moneda nativa o principal en Cardano. Esto significa que ada es la principal unidad de pago en Cardano.

Cardano también admite la creación de **tokens nativos**: activos digitales que se crean para fines específicos.

# Ada como criptomoneda de Cardano

Cada ledger de blockchain tiene su criptomoneda subyacente o moneda nativa. Ada es la moneda nativa o principal en Cardano. Esto significa que ada es la principal unidad de pago en Cardano.

Cardano también admite la creación de **tokens nativos**: activos digitales que se crean para fines específicos.

Por lo tanto los usuarios, desarrolladores y empresas pueden usar la cadena de bloques de Cardano para crear tokens que representen una huella de valor.

# Ada como criptomoneda de Cardano

Cada ledger de blockchain tiene su criptomoneda subyacente o moneda nativa. Ada es la moneda nativa o principal en Cardano. Esto significa que ada es la principal unidad de pago en Cardano.

Cardano también admite la creación de **tokens nativos**: activos digitales que se crean para fines específicos.

Por lo tanto los usuarios, desarrolladores y empresas pueden usar la cadena de bloques de Cardano para crear tokens que representen una huella de valor.

Un token puede ser **fungible** (intercambiable) o **no fungible** (único) y actuar como unidad de pago, recompensa, activo comercial o contenedor de información.

# Índice de contenidos

## 1 Introducción

- Blockchains, Criptomonedas y Smart contracts
- Cardano
- **ACTUS**
- Verificación formal

## 2 Escribiendo contratos ACTUS en Cardano

- Notación del estándar ACTUS
- Contratos en Cardano

## 3 Verificando propiedades en contratos en Marlowe

- El modelo de Marlowe
- Pruebas sencillas sobre contratos específicos
- Warnings en Auction

## 4 Conclusión

- Resumen
- Posibles temas de desarrollo futuro

## 5 Bibliografía



# Contratos Financieros

Los contratos financieros son acuerdos legales entre dos (o más) partes sobre el futuro intercambio de dinero. Dichos acuerdos legales se definen sin ambigüedades por medio de un conjunto de términos y lógica contractual.

# Contratos Financieros

Los contratos financieros son acuerdos legales entre dos (o más) partes sobre el futuro intercambio de dinero. Dichos acuerdos legales se definen sin ambigüedades por medio de un conjunto de términos y lógica contractual.

Como resultado, los mismos pueden describirse matemáticamente y representarse mediante algoritmos.

Los beneficios de representar contratos financieros de esta forma son múltiples:

Los beneficios de representar contratos financieros de esta forma son múltiples:

- Tradicionalmente, el procesamiento de transacciones ha sido un campo en el que se pueden lograr mejoras de eficiencia mediante la automatización de contratos.

Los beneficios de representar contratos financieros de esta forma son múltiples:

- Tradicionalmente, el procesamiento de transacciones ha sido un campo en el que se pueden lograr mejoras de eficiencia mediante la automatización de contratos.
- El análisis financiero se basa en la disponibilidad de representaciones computables de estos acuerdos, donde a menudo se utilizan aproximaciones analíticas.

Los beneficios de representar contratos financieros de esta forma son múltiples:

- Tradicionalmente, el procesamiento de transacciones ha sido un campo en el que se pueden lograr mejoras de eficiencia mediante la automatización de contratos.
- El análisis financiero se basa en la disponibilidad de representaciones computables de estos acuerdos, donde a menudo se utilizan aproximaciones analíticas. Recientemente, el auge de las blockchain, de contabilidad distribuida y los diversos casos de uso de los contratos inteligentes han abierto nuevas posibilidades para los contratos financieros digitales.

En general, el intercambio de flujos de efectivo entre partes sigue ciertos patrones.

En general, el intercambio de flujos de efectivo entre partes sigue ciertos patrones.

Un patrón típico es un contrato de préstamo de tipo *bullet*:



En general, el intercambio de flujos de efectivo entre partes sigue ciertos patrones.

Un patrón típico es un contrato de préstamo de tipo *bullet*:

*“Se entrega un monto de dinero inicial, a cambio de pagos de intereses cíclicos y la devolución del dinero inicial en el vencimiento del contrato.”*

En general, el intercambio de flujos de efectivo entre partes sigue ciertos patrones.

Un patrón típico es un contrato de préstamo de tipo *bullet*:

*“Se entrega un monto de dinero inicial, a cambio de pagos de intereses cíclicos y la devolución del dinero inicial en el vencimiento del contrato.”*

Si bien los pagos son fijos, existen muchas variantes que determinan cómo se programan y/o pagan los pagos de intereses cíclicos:

En general, el intercambio de flujos de efectivo entre partes sigue ciertos patrones.

Un patrón típico es un contrato de préstamo de tipo *bullet*:

*“Se entrega un monto de dinero inicial, a cambio de pagos de intereses cíclicos y la devolución del dinero inicial en el vencimiento del contrato.”*

Si bien los pagos son fijos, existen muchas variantes que determinan cómo se programan y/o pagan los pagos de intereses cíclicos:

- Los pagos de intereses pueden ser mensuales, anuales, mediante períodos arbitrarios.

En general, el intercambio de flujos de efectivo entre partes sigue ciertos patrones.

Un patrón típico es un contrato de préstamo de tipo *bullet*:

*“Se entrega un monto de dinero inicial, a cambio de pagos de intereses cíclicos y la devolución del dinero inicial en el vencimiento del contrato.”*

Si bien los pagos son fijos, existen muchas variantes que determinan cómo se programan y/o pagan los pagos de intereses cíclicos:

- Los pagos de intereses pueden ser mensuales, anuales, mediante períodos arbitrarios.
- Las tasas pueden ser de fijas o variables.

En general, el intercambio de flujos de efectivo entre partes sigue ciertos patrones.

Un patrón típico es un contrato de préstamo de tipo *bullet*:

*“Se entrega un monto de dinero inicial, a cambio de pagos de intereses cíclicos y la devolución del dinero inicial en el vencimiento del contrato.”*

Si bien los pagos son fijos, existen muchas variantes que determinan cómo se programan y/o pagan los pagos de intereses cíclicos:

- Los pagos de intereses pueden ser mensuales, anuales, mediante períodos arbitrarios.
- Las tasas pueden ser de fijas o variables.
- Pueden usarse diferentes métodos de cálculo de fracciones anuales o que no haya ningún interés.

# Índice de contenidos

## 1 Introducción

- Blockchains, Criptomonedas y Smart contracts
- Cardano
- ACTUS
- **Verificación formal**

## 2 Escribiendo contratos ACTUS en Cardano

- Notación del estándar ACTUS
- Contratos en Cardano

## 3 Verificando propiedades en contratos en Marlowe

- El modelo de Marlowe
- Pruebas sencillas sobre contratos específicos
- Warnings en Auction

## 4 Conclusión

- Resumen
- Posibles temas de desarrollo futuro

## 5 Bibliografía

# Concepto general, herramientas y metodologías

Los asistentes de pruebas formales son herramientas de software diseñadas para ayudar a sus usuarios a realizar pruebas, especialmente en cálculo lógico.

Por lo general, los llamamos asistentes de demostración o demostradores interactivos de teoremas.

# Concepto general, herramientas y metodologías

Los asistentes de pruebas formales son herramientas de software diseñadas para ayudar a sus usuarios a realizar pruebas, especialmente en cálculo lógico.

Por lo general, los llamamos asistentes de demostración o demostradores interactivos de teoremas.

La principal fortaleza de los asistentes de prueba es que ayudan a desarrollar pruebas altamente confiables e inequívocas de enunciados matemáticos, usando lógica precisa. Se pueden usar para probar resultados arbitrariamente avanzados, y no solo ejemplos simples.



# Algunos asistentes de pruebas

Hay una gran cantidad de asistentes de prueba en desarrollo o uso alrededor del mundo. A continuación presentamos una lista de los principales, clasificados por sus fundamentos lógicos:

# Algunos asistentes de pruebas

Hay una gran cantidad de asistentes de prueba en desarrollo o uso alrededor del mundo. A continuación presentamos una lista de los principales, clasificados por sus fundamentos lógicos:

- **Teoría de conjuntos:** Isabelle/ZF, Metamath, Mizar
- **Teoría simple de tipos:** HOL4, HOL Light, Isabelle/HOL
- **Teoría dependiente de tipos:** Agda, Coq, Lean, Matita, PVS
- **Lógica de primer orden, de tipo Lisp:** ACL2

# Índice de contenidos

- 1 **Introducción**
  - Blockchains, Criptomonedas y Smart contracts
  - Cardano
  - ACTUS
  - Verificación formal
- 2 **Escribiendo contratos ACTUS en Cardano**
  - **Notación del estándar ACTUS**
  - Contratos en Cardano
- 3 **Verificando propiedades en contratos en Marlowe**
  - El modelo de Marlowe
  - Pruebas sencillas sobre contratos específicos
  - Warnings en Auction
- 4 **Conclusión**
  - Resumen
  - Posibles temas de desarrollo futuro
- 5 **Bibliografía**

Antes de adentrarnos en la especificación de un contrato, es necesario poder entender algunos aspectos de la notación del mismo:

Antes de adentrarnos en la especificación de un contrato, es necesario poder entender algunos aspectos de la notación del mismo:

- **Atributos de contrato:** Representan los términos contractuales que definen el flujo de dinero en un contrato financiero.

Antes de adentrarnos en la especificación de un contrato, es necesario poder entender algunos aspectos de la notación del mismo:

- **Atributos de contrato:** Representan los términos contractuales que definen el flujo de dinero en un contrato financiero.
- **Starting date:**  $t_0$  representa la fecha de comienzo del contrato, y marca el instante en el cual las condiciones y estado del contrato están siendo representados.

Antes de adentrarnos en la especificación de un contrato, es necesario poder entender algunos aspectos de la notación del mismo:

- **Atributos de contrato:** Representan los términos contractuales que definen el flujo de dinero en un contrato financiero.
- **Starting date:**  $t_0$  representa la fecha de comienzo del contrato, y marca el instante en el cual las condiciones y estado del contrato están siendo representados.
- **Variables de estado:** Las variables de estado describen el estado de un contrato, para un tiempo determinado de su ciclo de vida. Algunos ejemplos de las mismas son:

Antes de adentrarnos en la especificación de un contrato, es necesario poder entender algunos aspectos de la notación del mismo:

- **Atributos de contrato:** Representan los términos contractuales que definen el flujo de dinero en un contrato financiero.
- **Starting date:**  $t_0$  representa la fecha de comienzo del contrato, y marca el instante en el cual las condiciones y estado del contrato están siendo representados.
- **Variables de estado:** Las variables de estado describen el estado de un contrato, para un tiempo determinado de su ciclo de vida. Algunos ejemplos de las mismas son:
  - Notional Principal.
  - Nominal Interest Rate.
  - Contract Performance.



Antes de adentrarnos en la especificación de un contrato, es necesario poder entender algunos aspectos de la notación del mismo:

- **Atributos de contrato:** Representan los términos contractuales que definen el flujo de dinero en un contrato financiero.
- **Starting date:**  $t_0$  representa la fecha de comienzo del contrato, y marca el instante en el cual las condiciones y estado del contrato están siendo representados.
- **Variables de estado:** Las variables de estado describen el estado de un contrato, para un tiempo determinado de su ciclo de vida. Algunos ejemplos de las mismas son:
  - Notional Principal.
  - Nominal Interest Rate.
  - Contract Performance.

En general, el 'estado' representa ciertos términos de un contrato que pueden cambiar a lo largo de su ciclo de ejecución, de acuerdo a eventos programados o no programados.

- **Eventos:** Un evento de contrato  $e_t^k$  se refiere a cualquier evento *programado o no programado* en un momento determinado  $t$  y de un tipo determinado  $k$ .

Los eventos del contrato marcan puntos específicos en el tiempo (durante la ejecución del mismo) en el que se intercambian flujos de efectivo o se actualizan los estados del contrato.

- **Eventos:** Un evento de contrato  $e_t^k$  se refiere a cualquier evento *programado o no programado* en un momento determinado  $t$  y de un tipo determinado  $k$ .

Los eventos del contrato marcan puntos específicos en el tiempo (durante la ejecución del mismo) en el que se intercambian flujos de efectivo o se actualizan los estados del contrato.

- **Secuencia de eventos:** Los eventos (de diferentes tipos) de un contrato pueden ocurrir en el mismo instante de tiempo  $t$ . Por lo tanto, se utiliza un indicador de secuencia de eventos que se puede encontrar para cada evento en el diccionario de eventos.

- **Eventos:** Un evento de contrato  $e_t^k$  se refiere a cualquier evento *programado o no programado* en un momento determinado  $t$  y de un tipo determinado  $k$ .  
Los eventos del contrato marcan puntos específicos en el tiempo (durante la ejecución del mismo) en el que se intercambian flujos de efectivo o se actualizan los estados del contrato.
- **Secuencia de eventos:** Los eventos (de diferentes tipos) de un contrato pueden ocurrir en el mismo instante de tiempo  $t$ . Por lo tanto, se utiliza un indicador de secuencia de eventos que se puede encontrar para cada evento en el diccionario de eventos.
- **Lifetime de contrato:** La vida útil de un contrato ACTUS es el período de tiempo de su existencia, desde la perspectiva del usuario que analiza. Se puede analizar un contrato ACTUS en términos de estado actual y flujos de efectivo futuros para cada punto del tiempo.

- **Funciones de transición de estado:** Dichas funciones, conocidas en Inglés como '*State Transition Functions*' (STF) definen la transición de las variables de estado desde el *pre-evento* hacia el *post-evento*, cuando un cierto evento  $e_t^k$  ocurre. Esto provoca que el *pre-evento* y *post-evento* reciban la notación de  $t^-$  y  $t^+$  respectivamente.

- **Funciones de transición de estado:** Dichas funciones, conocidas en Inglés como '*State Transition Functions*' (STF) definen la transición de las variables de estado desde el *pre-evento* hacia el *post-evento*, cuando un cierto evento  $e_t^k$  ocurre. Esto provoca que el *pre-evento* y *post-evento* reciban la notación de  $t^-$  y  $t^+$  respectivamente.

Estas funciones son específicas para un tipo de evento y contrato. Las mismas son escritas de acuerdo al siguiente formato

**STF\_[event type]\_[contract type]()**, donde [event type] y [contract type] hacen alusión al tipo de evento y contrato al cual la STF pertenece.

- **Funciones de transición de estado:** Dichas funciones, conocidas en Inglés como '*State Transition Functions*' (STF) definen la transición de las variables de estado desde el *pre-evento* hacia el *post-evento*, cuando un cierto evento  $e_t^k$  ocurre. Esto provoca que el *pre-evento* y *post-evento* reciban la notación de  $t^-$  y  $t^+$  respectivamente.

Estas funciones son específicas para un tipo de evento y contrato. Las mismas son escritas de acuerdo al siguiente formato

**STF\_[event type]\_[contract type]()**, donde [event type] y [contract type] hacen alusión al tipo de evento y contrato al cual la STF pertenece.

Por ejemplo: La STF para un evento de tipo IP en el contrato PAM se escribe como STF\_IP\_PAM () y modifica (entre otras) a la variable **lpac** desde el pre-evento **lpac** <sub>$t^-$</sub>  al post-evento **lpac** <sub>$t^+$</sub> .

- **Funciones de pago:** Las funciones de pago, o *Payoff Functions* (POF) definen como el flujo de dinero  $c \in \mathbb{R}$  ocurre para un determinado evento  $e_t^k$ . El mismo es obtenido del estado actual y los términos del contrato. Si fuera necesario, el flujo de dinero puede ser indexado con el tiempo del evento:  $c_t$ .



- **Funciones de pago:** Las funciones de pago, o *Payoff Functions* (POF) definen como el flujo de dinero  $c \in \mathbb{R}$  ocurre para un determinado evento  $e_t^k$ . El mismo es obtenido del estado actual y los términos del contrato. Si fuera necesario, el flujo de dinero puede ser indexado con el tiempo del evento:  $c_t$ .

Las funciones de pago (de forma análoga a las STF), son específicas para un tipo de evento y contrato, y su notación es la siguiente:

**POF\_[event type]\_[contract type] ()**, donde [event type] y [contract type] hacen alusión al tipo de evento y contrato al cual la POF pertenece.

- **Funciones de pago:** Las funciones de pago, o *Payoff Functions* (POF) definen como el flujo de dinero  $c \in \mathbb{R}$  ocurre para un determinado evento  $e_t^k$ . El mismo es obtenido del estado actual y los términos del contrato. Si fuera necesario, el flujo de dinero puede ser indexado con el tiempo del evento:  $c_t$ .

Las funciones de pago (de forma análoga a las STF), son específicas para un tipo de evento y contrato, y su notación es la siguiente:

**POF\_[event type]\_[contract type] ()**, donde [event type] y [contract type] hacen alusión al tipo de evento y contrato al cual la POF pertenece.

Por ejemplo: La POF para un evento de tipo IP  $e_t^{IP}$  en el contrato PAM se escribe como POF\_IP\_PAM().

# Índice de contenidos

- 1 **Introducción**
  - Blockchains, Criptomonedas y Smart contracts
  - Cardano
  - ACTUS
  - Verificación formal
- 2 **Escribiendo contratos ACTUS en Cardano**
  - Notación del estándar ACTUS
  - **Contratos en Cardano**
- 3 **Verificando propiedades en contratos en Marlowe**
  - El modelo de Marlowe
  - Pruebas sencillas sobre contratos específicos
  - Warnings en Auction
- 4 **Conclusión**
  - Resumen
  - Posibles temas de desarrollo futuro
- 5 **Bibliografía**

# Introducción

En esta sección veremos como se desarrolló la escritura de tres contratos ACTUS para la blockchain Cardano, bajo la supervisión de IOHK.

Cabe destacar que durante esta tarea tuve la colaboración de Yves Hauser<sup>1</sup>, con quien conversamos sobre decisiones de diseño e implementación de los contratos correspondientes. Yves fue el responsable de integrar mis cambios a la rama *master* del repositorio de `marlowe-cardano` [IOHK, 2016].

---

<sup>1</sup><https://iohk.io/en/team/yves-hauser>

# Estructura del proyecto ACTUS en Cardano

A grandes rasgos, la estructura del generador de contratos ACTUS tiene las siguientes partes:

- **Domain:** El mismo está conformado por los archivos que modelan el dominio de los contratos ACTUS.

# Estructura del proyecto ACTUS en Cardano

A grandes rasgos, la estructura del generador de contratos ACTUS tiene las siguientes partes:

- **Domain:** El mismo está conformado por los archivos que modelan el dominio de los contratos ACTUS.

Entre ellos se pueden destacar:

```
└─ Domain
   └─ BusinessEvents.hs
   └─ ContractState.hs
   └─ ContractTerms.hs
   └─ Ops.hs
   └─ Schedule.hs
```

- **Generator:** En este directorio se implementan los diferentes generadores y compatibilidad hacia el lenguaje Marlowe.

- **Generator:** En este directorio se implementan los diferentes generadores y compatibilidad hacia el lenguaje Marlowe.

La estructura de dicho directorio es la siguiente:

```
Generator
├── Analysis.hs
├── Generator.hs
├── GeneratorFs.hs
├── GeneratorStatic.hs
└── MarloweCompat.hs
```



- **Model:** En este directorio se encuentran los archivos que modelan la lógica expuesta por el estándar ACTUS, tales como el *scheduling*, la inicialización de variables de estado y funciones de transición de estado y de pago.

- **Model:** En este directorio se encuentran los archivos que modelan la lógica expuesta por el estándar ACTUS, tales como el *scheduling*, la inicialización de variables de estado y funciones de transición de estado y de pago.

```
Model
├── Applicability.hs
├── ContractSchedule.hs
├── Payoff.hs
├── StateInitialization.hs
└── StateTransition.hs
```

- **Model:** En este directorio se encuentran los archivos que modelan la lógica expuesta por el estándar ACTUS, tales como el *scheduling*, la inicialización de variables de estado y funciones de transición de estado y de pago.

```
Model
├── Applicability.hs
├── ContractSchedule.hs
├── Payoff.hs
├── StateInitialization.hs
└── StateTransition.hs
```

- **Utility:** En este directorio se encuentran algunos archivos con funciones que se utilizan para aislar la lógica del cálculo de fechas, que suele tornarse complejo y repetitivo durante los contratos.

# Índice de contenidos

- 1 **Introducción**
  - Blockchains, Criptomonedas y Smart contracts
  - Cardano
  - ACTUS
  - Verificación formal
- 2 **Escribiendo contratos ACTUS en Cardano**
  - Notación del estándar ACTUS
  - Contratos en Cardano
- 3 **Verificando propiedades en contratos en Marlowe**
  - **El modelo de Marlowe**
  - Pruebas sencillas sobre contratos específicos
  - Warnings en Auction
- 4 **Conclusión**
  - Resumen
  - Posibles temas de desarrollo futuro
- 5 **Bibliografía**

# Índice de contenidos

- 1 **Introducción**
  - Blockchains, Criptomonedas y Smart contracts
  - Cardano
  - ACTUS
  - Verificación formal
- 2 **Escribiendo contratos ACTUS en Cardano**
  - Notación del estándar ACTUS
  - Contratos en Cardano
- 3 **Verificando propiedades en contratos en Marlowe**
  - El modelo de Marlowe
  - **Pruebas sencillas sobre contratos específicos**
  - Warnings en Auction
- 4 **Conclusión**
  - Resumen
  - Posibles temas de desarrollo futuro
- 5 **Bibliografía**

# Índice de contenidos

- 1 **Introducción**
  - Blockchains, Criptomonedas y Smart contracts
  - Cardano
  - ACTUS
  - Verificación formal
- 2 **Escribiendo contratos ACTUS en Cardano**
  - Notación del estándar ACTUS
  - Contratos en Cardano
- 3 **Verificando propiedades en contratos en Marlowe**
  - El modelo de Marlowe
  - Pruebas sencillas sobre contratos específicos
  - **Warnings en Auction**
- 4 **Conclusión**
  - Resumen
  - Posibles temas de desarrollo futuro
- 5 **Bibliografía**

# Índice de contenidos


- 1 **Introducción**
  - Blockchains, Criptomonedas y Smart contracts
  - Cardano
  - ACTUS
  - Verificación formal
- 2 **Escribiendo contratos ACTUS en Cardano**
  - Notación del estándar ACTUS
  - Contratos en Cardano
- 3 **Verificando propiedades en contratos en Marlowe**
  - El modelo de Marlowe
  - Pruebas sencillas sobre contratos específicos
  - Warnings en Auction
- 4 **Conclusión**
  - **Resumen**
  - Posibles temas de desarrollo futuro
- 5 **Bibliografía**


# Índice de contenidos

- 1 **Introducción**
  - Blockchains, Criptomonedas y Smart contracts
  - Cardano
  - ACTUS
  - Verificación formal
- 2 **Escribiendo contratos ACTUS en Cardano**
  - Notación del estándar ACTUS
  - Contratos en Cardano
- 3 **Verificando propiedades en contratos en Marlowe**
  - El modelo de Marlowe
  - Pruebas sencillas sobre contratos específicos
  - Warnings en Auction
- 4 **Conclusión**
  - Resumen
  - Posibles temas de desarrollo futuro
- 5 **Bibliografía**



# Bibliografía I

 Brünjes, L. and Vinogradova, P. (2019).  
*Plutus: Writing reliable smart contracts.*  
IOHK.

 IOHK (2016).  
Marlowe cardano repository.  
<https://github.com/input-output-hk/marlowe-cardano>.  
Online; accessed 26 May 2022.