Tecnoloxías de rexistro distribuído e Blockchain

Máster Inter-Universitario en Ciberseguridad (MUNICS)
Universidade da Coruña (UDC) y Universidade de Vigo (UVigo)
Curso 2024-2025

Ejercicio 6: Contrato inteligente: Alquiler de pisos

Alexis Bernárdez Hermida, Nuria Codesido Iglesias

1. Análisis y definición del escenario.

El contrato *AlquilerPisos* está diseñado para gestionar alquileres de viviendas en un entorno descentralizado. En este escenario, las partes involucradas interactúan directamente a través del contrato inteligente, eliminando la necesidad de intermediarios.

Es importante señalar que el contrato establece varios plazos de 30 minutos para la ejecución de ciertas funciones, como la aceptación de nuevas propuestas y el pago del alquiler. Estos plazos garantizan que los usuarios puedan comprobar de manera efectiva la funcionalidad del contrato.

La implementación de este contrato en la blockchain de Ethereum busca cumplir con una serie de **requisitos no funcionales**:

- **RNF- 01**. Inmutabilidad: Los registros de transacciones, como el pago del alquiler o la devolución de fianzas, no podrán ser alterados.
- **RNF- 02**. Transparencia y visibilidad: Las partes involucradas, el arrendador y arrendatario, tienen acceso completo a la información relevante del contrato.
- **RNF- 03**. Descentralización: Al ejecutarse en una blockchain pública, el contrato no depende de una entidad centralizada.
- **RNF- 04**. Automatización: La ejecución automática de procesos, como el pago de alquiler y la devolución de fianzas, elimina la necesidad de intermediarios y garantiza que se cumplan sin intervención manual.

A continuación se especifican los **requisitos funcionales** del contrato inteligente:

- **RF- 01**. Proponer contrato: Permite al arrendador proponer nuevas condiciones de contrato, duración y precio, antes de los 30 minutos del final del contrato actual, siempre que no exista otra propuesta pendiente. Una vez propuesta, el arrendatario debe aceptarla dentro de un plazo de 30 minutos.

- **RF- 02**. Aceptar contrato: El arrendatario podrá aceptar la propuesta dentro del período especificado (30 minutos), activando así las nuevas condiciones del contrato.

- **RF- 03**. Pago del alquiler: Si el contrato está activo y el plazo del pago está dentro de los 30 minutos, el arrendatario podrá realizar el pago del alquiler. En el caso de que sea la primera vez que se realiza el pago, el arrendatario deberá pagar tanto el alquiler como la fianza. Si el arrendatario no realiza el pago a tiempo, se aplicará una penalización por impago.

- **RF- 04**. Penalizar por impago: Si el arrendatario no realiza el pago del alquiler a tiempo, el arrendador podrá establecer una penalización. Esta penalización se calcula en función del tiempo transcurrido desde la fecha de vencimiento del pago y se asigna una cantidad específica como sanción por el incumplimiento.

- **RF- 05**. Pagar penalización: El arrendatario podrá pagar la penalización establecida si se ha excedido el plazo de pago. Esta función asegura que se realice el pago correcto de la penalización y que el arrendador reciba la cantidad correspondiente.

- **RF- 06**. Finalizar contrato: El arrendador puede finalizar el contrato siempre que esté activo y el saldo del contrato sea suficiente para realizar las devoluciones pertinentes. Las razones para finalizar el contrato son las siguientes:
  - Mutuo acuerdo: En este caso, se devuelve la fianza al completo.
  - Incumplimiento, daños: En este caso, se devuelve la fianza reducida.
  - Desalojo: En este caso, no se devuelve la fianza y se aplica una penalización del 10% de la fianza.
  - Cambio arrendador: Permite cambiar al arrendador del contrato.

- **RF- 07**. Devolver fianza: Al finalizar el contrato, se comprueba que el contrato no haya finalizado, que la fianza no haya sido devuelta y que la cantidad a retener por daños o incumplimientos no exceda la cantidad de la fianza estipulada. Además, se comprobará que el contrato tenga suficiente saldo para realizar la devolución. Si se cumplen estas condiciones, el arrendador puede devolver la fianza al arrendatario, descontando las deducciones correspondientes según lo establecido.

2. Diseño.

2.1. Definir un caso de uso (Pagar el alquiler).

Precondiciones:

- Contrato activo: El contrato de arrendamiento debe estar activo.

- La función debe ser invocada por el user: arrendatario.
- Fecha vencimiento: El pago debe realizarse antes de la fecha límite definida.
- El arrendatario debe tener suficiente saldo en su cuenta para realizar el pago.
- En el caso de que sea el primer pago de alquiler, se debe realizar un pago que incluya tanto el alquiler mensual como la fianza.

Flujo:

1. El arrendatario ejecuta la función *pagarAlquiler*.
2. El contrato verifica que el pago se realiza antes de la fecha de vencimiento y que está activo.
3. El contrato transfiere la cantidad correspondiente al arrendador.
4. Se actualiza la fecha de vencimiento sumándole 30 minutos.
5. Se emite un evento que indica que el pago ha sido realizado.

Postcondiciones:

- La cantidad correspondiente al alquiler se transfiere al arrendador.
- Se registra el evento de pago que confirma la transacción.
- La fecha de vencimiento para el próximo pago del alquiler se actualiza correctamente.

2.2. Definir el contenido del contrato inteligente.

2.2.1. Datos.

- *arrendador*: Almacena la dirección del arrendador (propietario de la vivienda).
- *arrendatario*: Almacena la dirección del arrendatario (inquilino de la vivienda).
- *precioAlquiler*: Almacena el precio del alquiler.
- *fianzaAlquiler*: Almacena el precio de la fianza.
- *duracionContrato*: Almacena la duración del contrato de alquiler en días.
- *fechaInicioContrato*: Registra la fecha de inicio del contrato.
- *fechaVencimientoPago*: Registra la fecha límite para el pago del alquiler.
- *fianzaDevuelta*: Indica si la fianza ha sido devuelta al arrendatario.
- *contratoActivo*: Indica si el contrato de alquiler está actualmente activo.
- *pagoPrimerMes*: Indica si el pago del primer mes (que incluye la fianza) ha sido realizado.
- *penalizacion*: Almacena la cantidad de penalización impuesta por impago.
- *motivosFinalizacion*: Enumera los diferentes motivos por los cuales se puede finalizar el contrato: `INCUMPLIMIENTO`, `MUTUOACUERDO`, `NORENOVACION`, `DESALOJO`, `DAMAGE`, `CAMBIOARRENDADOR`, `RENUNCIA`.
- *nuevaDuracionPropuesta*: Almacena la nueva duración del contrato que el arrendador propone.
- *nuevoPrecioPropuesta*: Almacena el nuevo precio del contrato que el arrendador propone.

- *propuestaPendiente*: Indica si hay una propuesta de contrato pendiente de aceptación por parte del arrendatario.
- *plazoPropuesta*: Almacena el momento en el que se realizó la propuesta.

2.2.2. Eventos

- *pagoRealizado*: Se emite cuando el arrendatario realiza un pago del alquiler.
- *devolucionFianza*: Se emite cuando la fianza es devuelta al arrendatario.
- *propuestaRenovacion*: Se emite cuando el arrendador propone una nueva renovación del contrato.
- *acuerdoAlcanzado*: Se emite cuando el arrendatario acepta la propuesta de contrato.
- *contratoFinalizado*: Se emite al finalizar el contrato, especificando el motivo.
- *penalizacionEstablecida*: Se emite cuando se establece una penalización por impago.
- *penalizacionPagada*: Se emite cuando el arrendatario paga la penalización.

2.2.3. Modificadores

- *soloArrendador*: Restringe el acceso a las funciones a sólo el arrendador.
- *soloArrendatario*: Restringe el acceso a las funciones a sólo el arrendatario.

2.2.4. Funciones

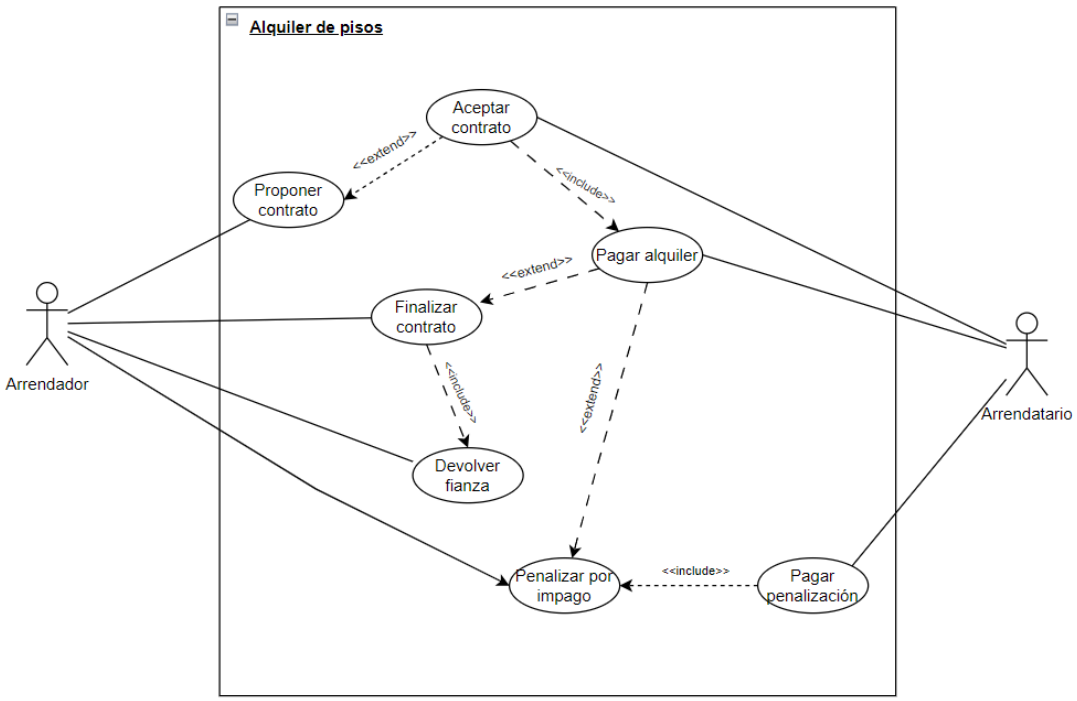
- *Constructor*: Constructor del contrato que se ejecuta una vez al ser desplegado. Inicializa los parámetros esenciales del contrato, tales como el arrendador y el arrendatario, el precio y la fianza del alquiler, y la duración del contrato. Además, activa el contrato estableciendo la propiedad contratoActivo en true. También calcula y establece la fecha de inicio del contrato y la fecha de vencimiento del primer pago. Esto asegura que todas las condiciones iniciales estén definidas y que el contrato esté preparado para que el arrendatario realice el pago.
- *pagarAlquiler*: Permite al arrendatario realizar el pago del alquiler.
- *devolverFianza*: Permite al arrendador devolver la fianza al arrendatario al final del contrato.
- *proponerContrato*: Permite al arrendador proponer nuevas condiciones para el contrato de alquiler.
- *aceptarContrato*: Permite al arrendatario aceptar la propuesta de renovación del contrato.
- *finalizarContrato*: Permite al arrendador finalizar el contrato por diferentes motivos. Dependiendo del motivo, se realizan acciones específicas, como devolver o retener la fianza.
- *penalizarPorImpago*: Permite al arrendador establecer una penalización por impago si se ha excedido la fecha de vencimiento del pago.
- *pagarPenalizacion*: Permite al arrendatario pagar la penalización por impago.

- *motivoToString*: convierte un valor de un enumerado llamado *motivosFinalizacion* en su representación en forma de cadena de texto (string).

2.6. Actores

Este contrato incluye dos partes involucradas: el **arrendador** y el **arrendatario**. El arrendador es el propietario de la vivienda que establece el alquiler. Por otro lado, el arrendatario es la persona que alquila la propiedad.

Diagrama de casos de uso



3. Implementación.

Para la implementación de este contrato inteligente se ha empleado un fichero: *AlquilerPisos.sol*

4. Pruebas.

Las pruebas del contrato inteligente se realizaron de forma manual con la ayuda de la herramienta **Remix - Ethereum IDE**.