Logotipo

Descripción generada automáticamente

**Sistemas de Información Empresariales**

Curso 2022-23

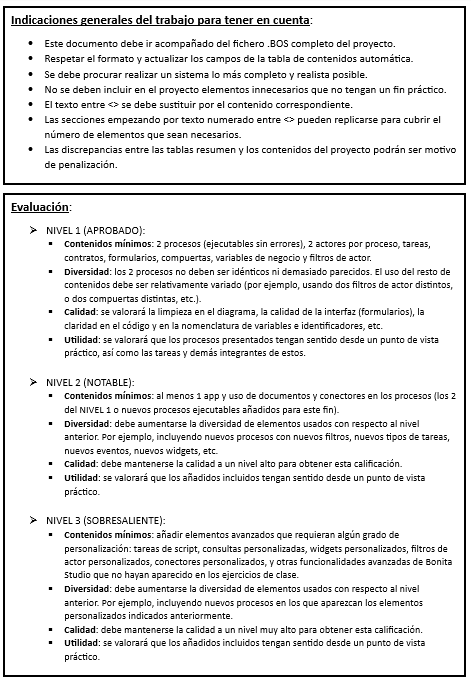
Trabajo sobre Bonita BPM

**Trabajo en equipo asociado**: HÉRCULES SEGURIDAD

**Integrantes**:

1. AVILÉS GIL, ELENA (ELEAVIGIL)
2. AGUILAR MARTÍN, ALBERTO (ALBAGUMAR)
3. GRIJUELA MUÑOZ, MANUEL (MANGRIMUN)
4. OGALLA LLAVERO, MARÍA DE LA PAZ (MARLLAOGA)

domingo, 14 de mayo de 2023



Contenido

[Descripción general](#_heading=h.gjdgxs)

[Nivel al que se opta](#_heading=h.30j0zll)

[Elementos destacados incluidos en el proyecto](#_heading=h.1fob9te)

[Comentarios](#_heading=h.3znysh7)

[Contexto](#_heading=h.2et92p0)

Añadir nuevo cliente

Creación de sistemas

Revisión de sistemas Instalados

[Organización](#_heading=h.3dy6vkm)

[Descripción](#_heading=h.1t3h5sf)

[Estructura](#_heading=h.4d34og8)

alberto.aguilar

[Características](#_heading=h.2s8eyo1)

[Business Data Model (BDM)](#_heading=h.3rdcrjn)

[Descripción](#_heading=h.26in1rg)

[Características](#_heading=h.lnxbz9)

Cliente

Sistema

[Procesos](#_heading=h.1ksv4uv)

NuevoCliente

[Diagrama BPMN](#_heading=h.2jxsxqh)

[Características](#_heading=h.z337ya)

nuevoClienteForm

revisarClienteForm

llamadaForm

presupuestoForm

Revisión

[Diagrama BPMN](#_heading=h.247tkwqr27th)

Características

[visValidacion](#_heading=h.n9t4pnf24pie)

CrearSistema

[Diagrama BPMN](#_heading=h.247tkwqr27th)

Características

nuevoSistemaForm

revisarSistemaForm

asignarDateForm

# Descripción general

## Nivel al que se opta

NIVEL: 3

### Elementos destacados incluidos en el proyecto

*Añade únicamente las aportaciones que no aparezca más adelante en las secciones posteriores*

| Páginas | carteraClientes: Permite consultar y añadir clientes a la base de datos |
| --- | --- |
| Aplicaciones | Cartera de clientes: Aplicación dentro de la cual se encuentra la página anterior |
| Widgets personalizados |  |
| Scripts (Groovy/Java) | Varios scripts dentro del proceso revisión encargados de procesar los resultados de las consultas a la base de datos externa, los cuales mapean a cada tipo de sistema el número de sistemas de tal tipo que se encuentran instalados |
| Código Javascript |  |
| Consultas personalizadas |  |
|  |  |
| Filtros de actor personalizados |  |
| Conectores personalizados | -Consulta Odoo: realiza una consulta a la base de datos de Odoo (más elaborado en el proceso Revisión)  -Insert Odoo: realiza un insert en la base de datos de Odoo, también dentro del proceso revisión, de una serie de datos procesados durante el proceso |
| Otros | A partir de un formulario y una plantilla de documento hemos podido enviar un presupuesto relleno al cliente (usamos dos conectores, uno de conversor a documento y otro de correo) |

### Comentarios

Hemos hecho la mayoría de requisitos de nivel 2 aunque no todos pero también hemos hecho algunos de los requisitos de nivel 3.

## Contexto

Hemos diseñado este bpm para la empresa “Hércules Seguridad”, una empresa que se dedica a la instalación y mantenimiento de sistemas de videovigilancia, anti-robos y contraincendios.

La implementación se hace con el objetivo de hacer más sencillos los procesos de crear clientes nuevos e instalar nuevos sistemas, así como integrarlo con todo lo realizado en el trabajo de odoo para optimizar los procesos administrativos de la empresa.

## Añadir un nuevo cliente

Este proceso trata de crear un nuevo cliente a partir de un lead. Al crearse el lead, se introducen ciertos datos del usuario, se contacta con él para ver si sigue interesado y, en caso afirmativo, se le envía un presupuesto por correo electrónico. Es una forma muy eficaz de captar nuevos clientes.

## CREACIÓN DE SISTEMAS

Se permiten crear nuevos sistemas por un propio técnico, ser aceptados o rechazados y asignarles una fecha en la que realizar la instalación, así como añadirlo a la lista de sistemas que tiene cada cliente. Para ello simplemente se debe ejecutar el proceso CrearSistema.

## REVISIÓN DE SISTEMAS INSTALADOS

Nuestro sistema permite compatibilizar con nuestro trabajo de teoría, realizado en Odoo. Dentro de Odoo existe una tabla llamada sistemas que almacena todos los sistemas instalados hasta ese momento, así como el tipo de sistema (Sistema anti-incendios, Videovigilancia,...) Cuando se quiera realizar una revisión de estos datos, se inicia el proceso “revisión”, en el que se entra más en detalle más adelante en el trabajo.

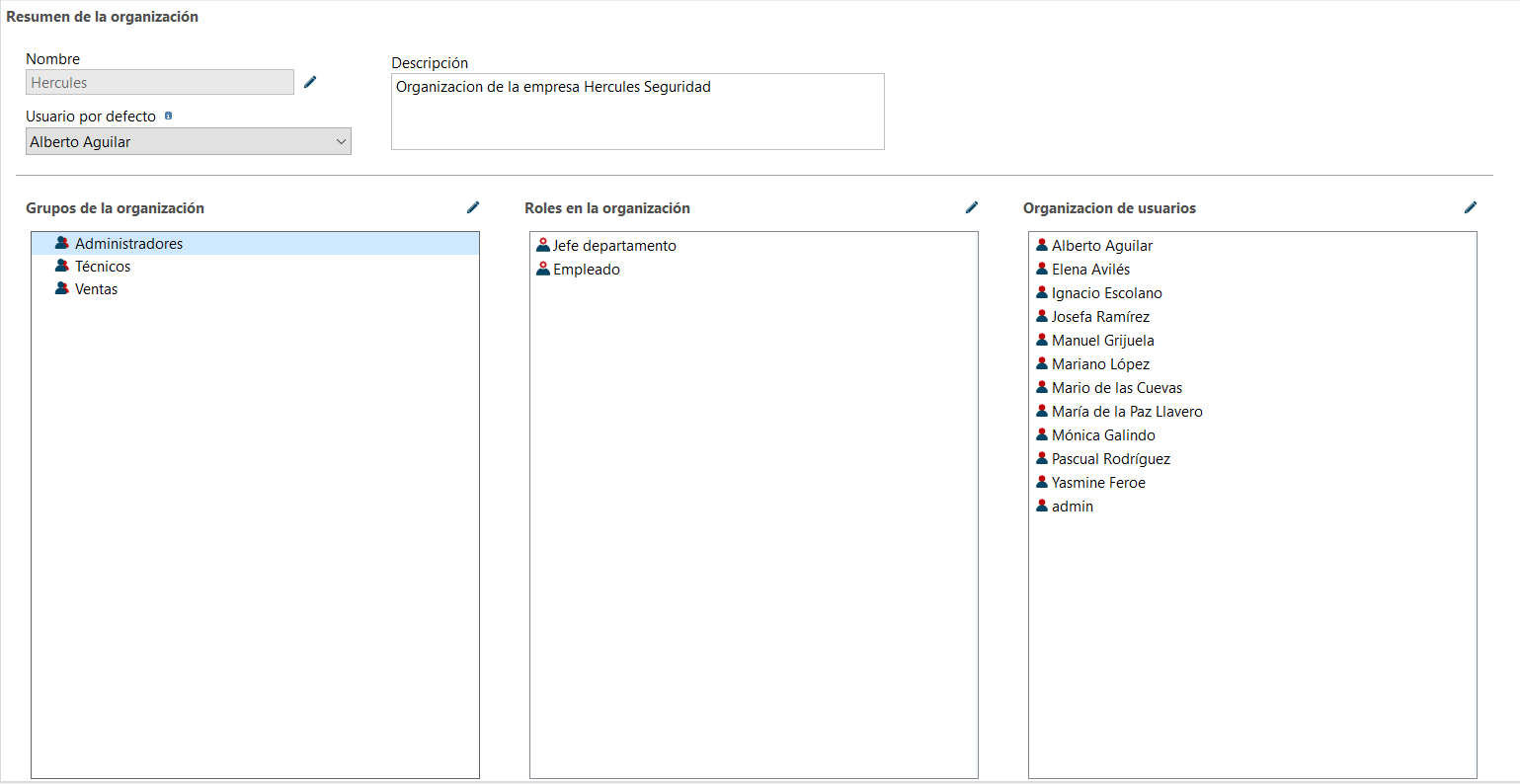
Se creará una tarea para un administrador de la empresa en la que le aparecerá una gráfica con las proporciones de cada tipo de sistema en función del número de veces que aparece. Si todo es correcto, el usuario confirma la revisión y el resultado de la misma se almacena dentro de la base de datos de Odoo de nuevo en otra tabla personalizada.

# Organización

## Descripción

Nuestra organización posee la misma estructura que la empresa en la que nos basamos. Existen 3 departamentos: Administración, técnico y ventas. Dentro de los 3 departamentos anteriores existen 2 posibles roles: Jefe de departamento o empleado.

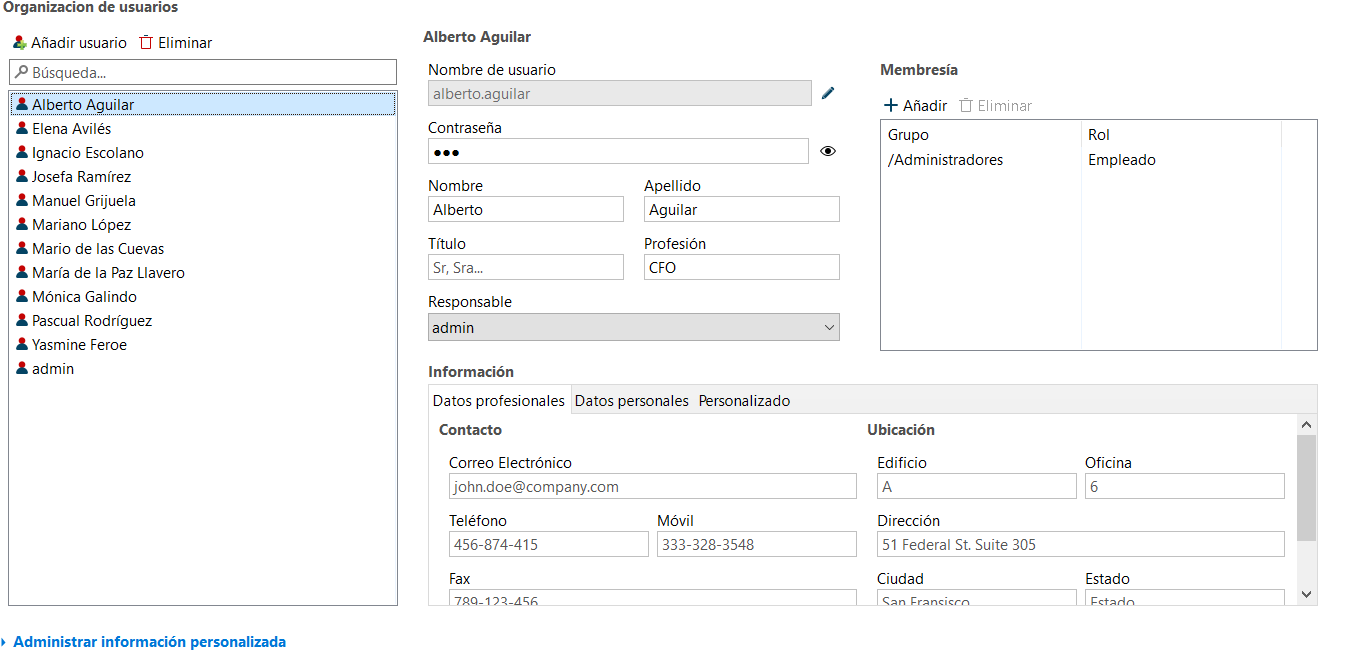
## Estructura



## Resumen de características

| Número de grupos | 3 |
| --- | --- |
| Número de roles | 2 |
| Número de usuarios | 12 |
| Número de membresías | 1 |

## alberto.aguilar



# Business Data Model (BDM)

## Descripción

Hemos añadido dos objetos distintos: Cliente y Sistema

Cliente: Guarda los datos de contacto de un cliente y una lista de sistemas asociados.

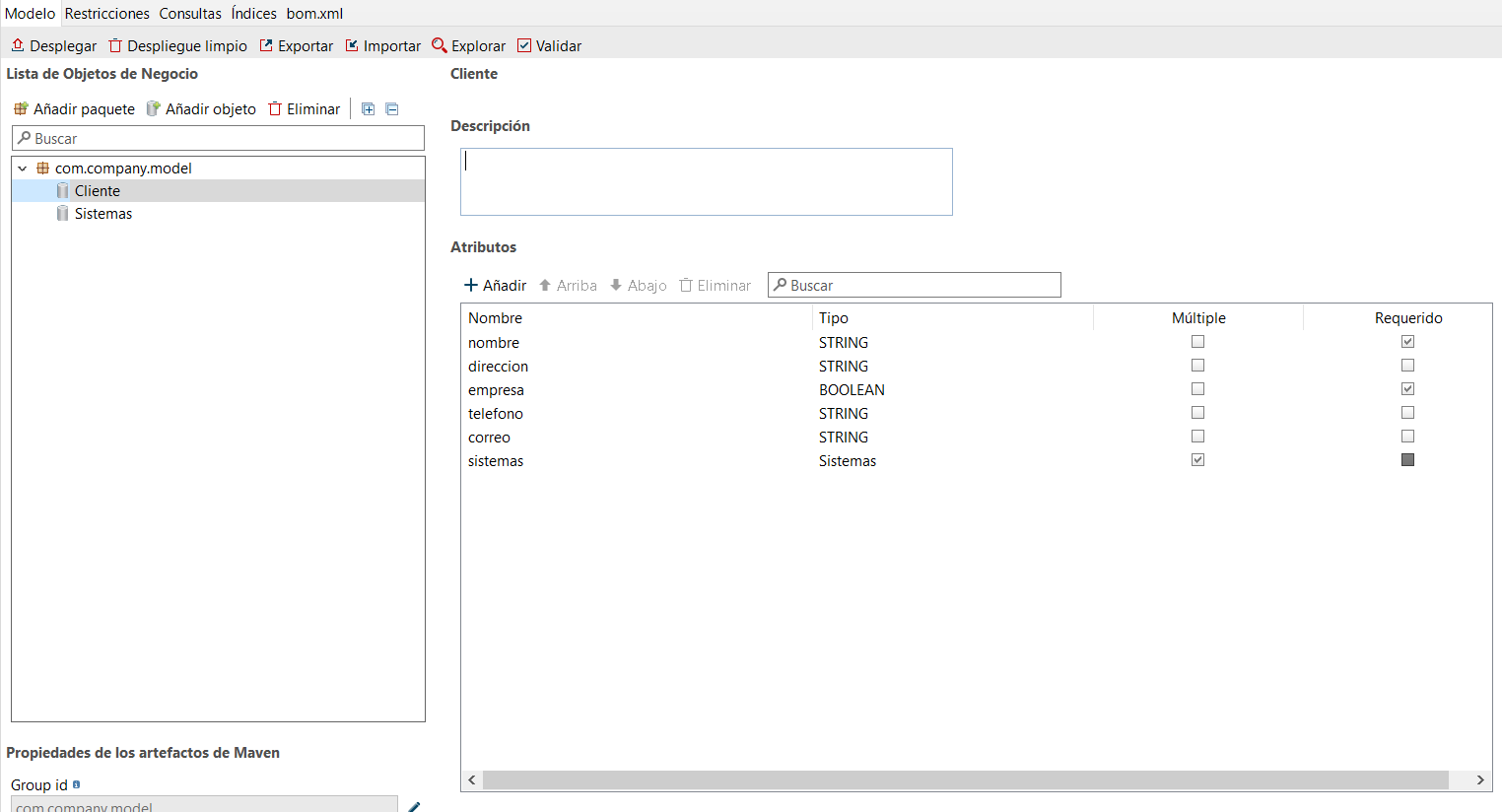
Sistema: Guarda cada sistema que se ha instalado (Videovigilancia, seguridad perimetral,...)

### Resumen de características

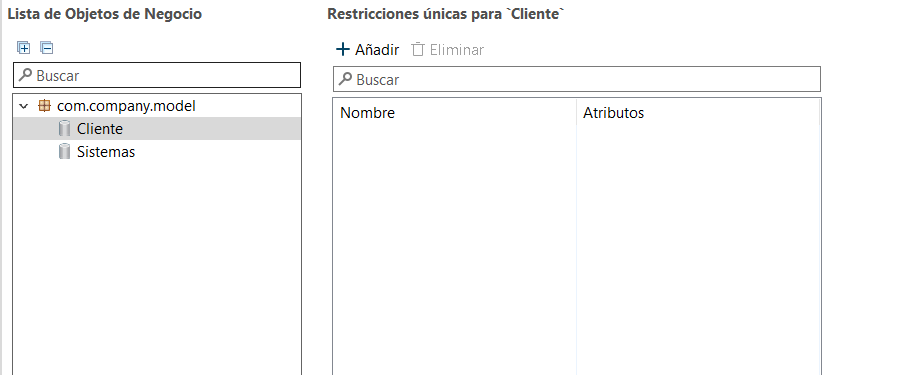
| Número de objetos | 2 |
| --- | --- |
| Group id utilizado |  |

### Cliente

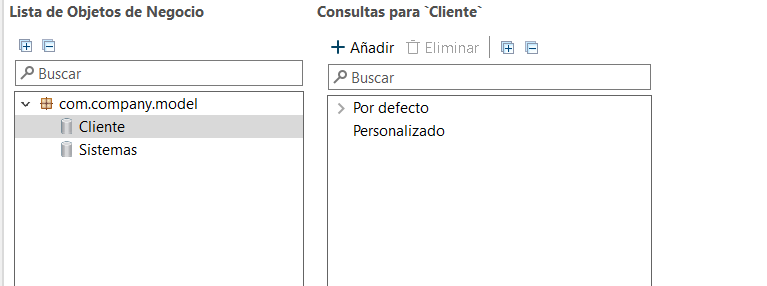
* Ejecución



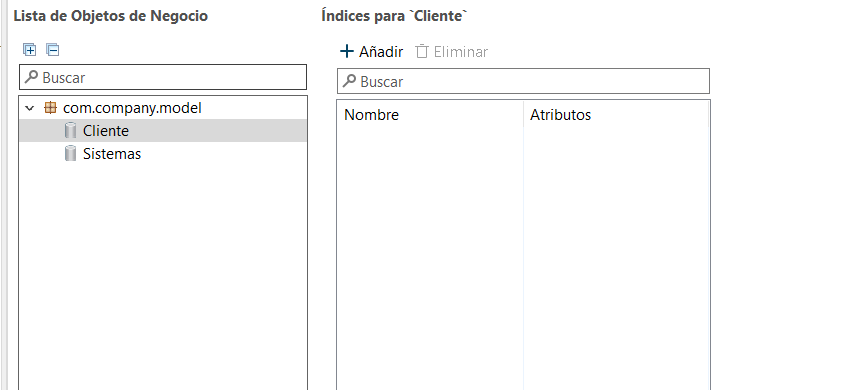
* Restricciones



* Consultas



* Índices

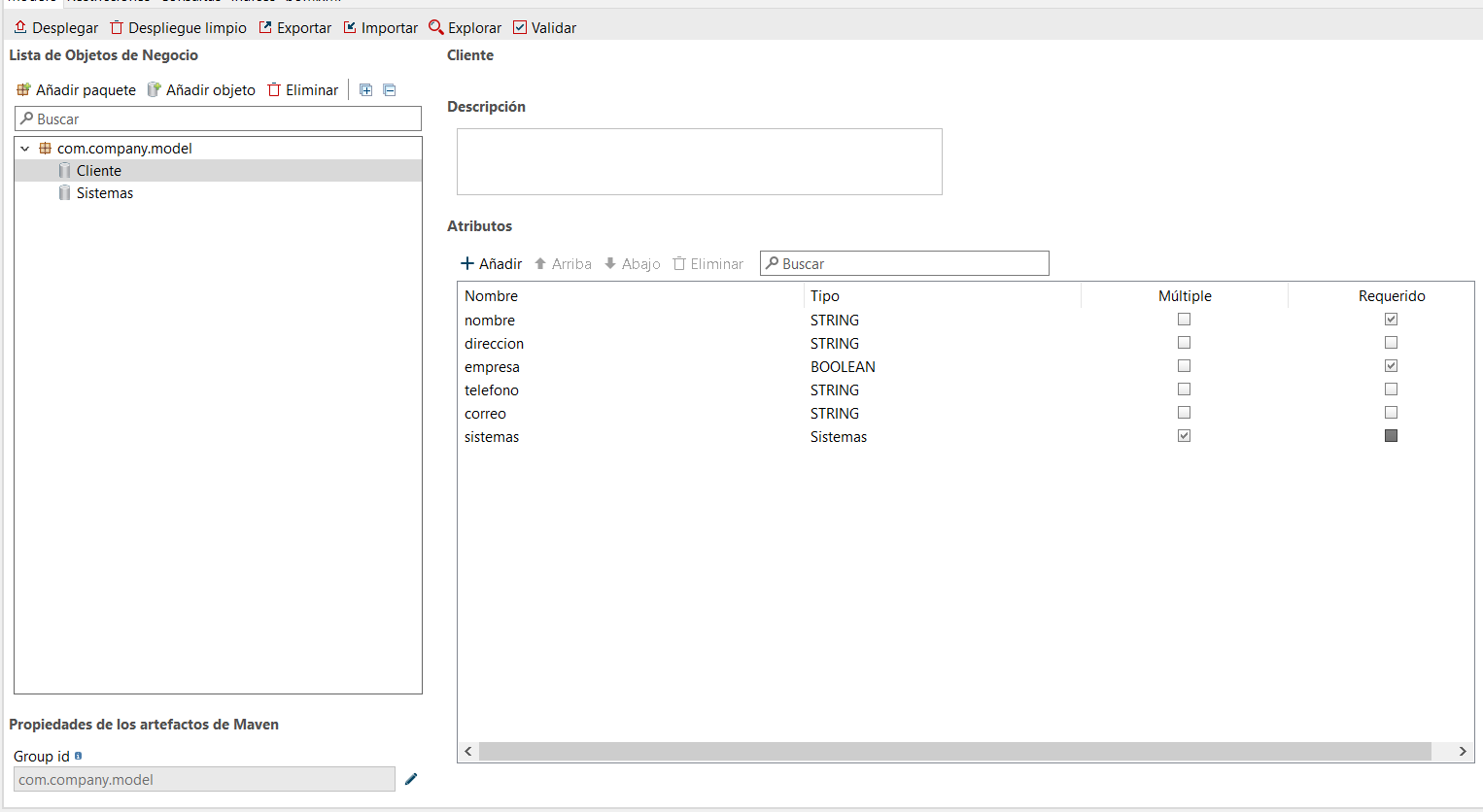


* Resumen de características del objeto de negocio

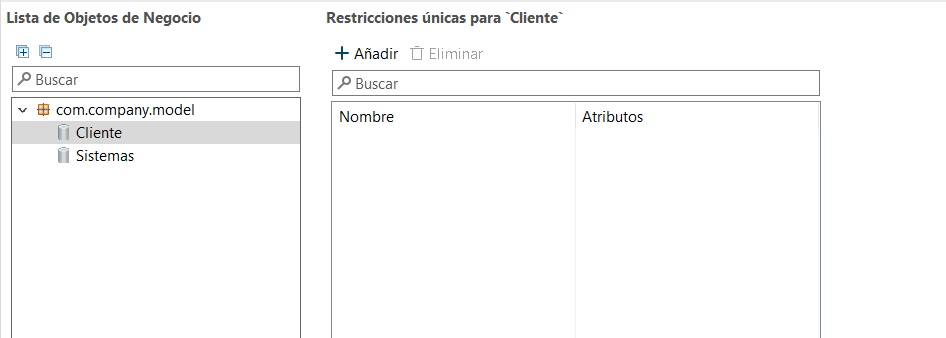
| Número de campos de texto | 4 |
| --- | --- |
| Número de campos numéricos |  |
| Número de campos booleanos | 1 |
| Número de campos relacionales |  |
| Número de campos múltiples | 1 |
| Número de restricciones |  |
| Número de consultas personalizadas |  |
| Número de índices |  |

### Sistema

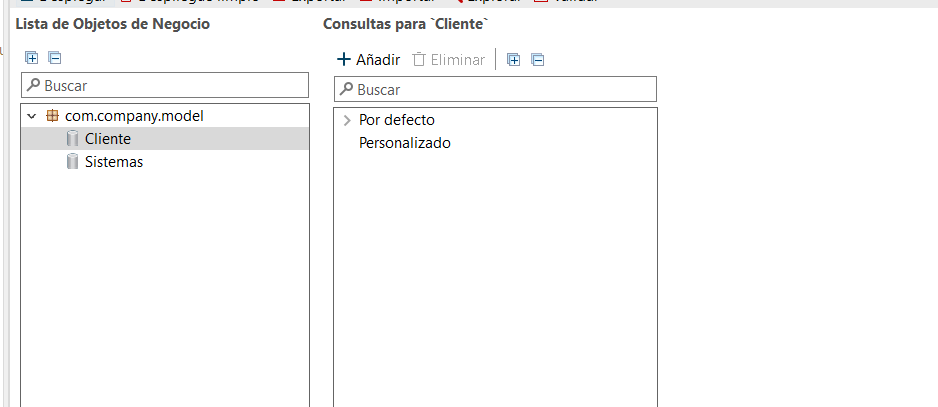
* Ejecución



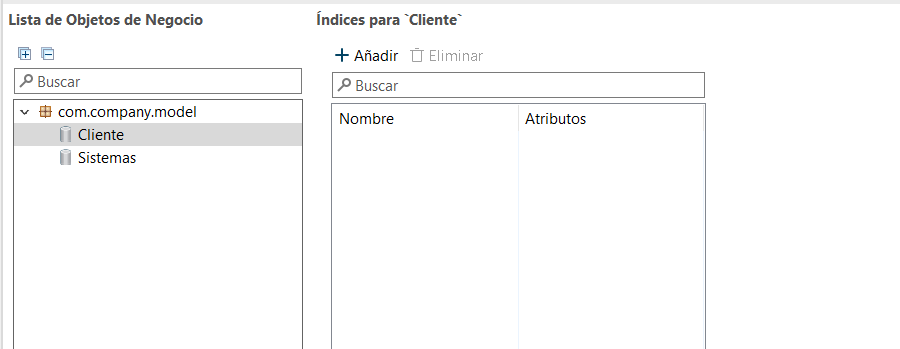
* Restricciones



* Consultas



* Índices



* Resumen de características del objeto de negocio

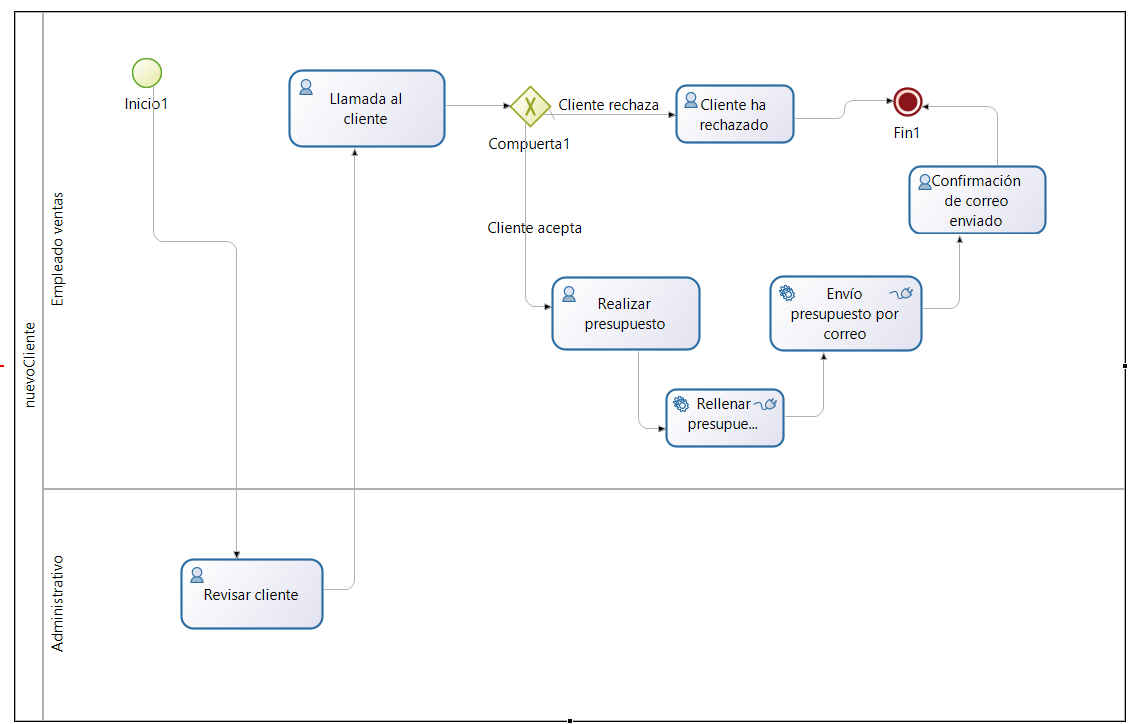
| Número de campos de texto | 3 |
| --- | --- |
| Número de campos numéricos | 1 |
| Número de campos booleanos | 1 |
| Número de campos relacionales | 1 |
| Número de campos múltiples |  |
| Número de restricciones |  |
| Número de consultas personalizadas |  |
| Número de índices |  |

# Procesos

## NuevoCliente

El proceso lo inicia un empleado de ventas que rellena su formulario de instanciación y esos datos le llegan a un administrador que confirma que los datos han llegado correctamente. Después, se produce una tarea en la que el vendedor que ha iniciado el proceso llama al cliente y confirma si quiere continuar con la venta. En caso negativo, se crea una tarea que avisa del rechazo y se acaba el proceso. En caso afirmativo, se crea un formulario con los datos a rellenar para crear un presupuesto. Una vez relleno, se crea una tarea automática en la que se usa un conector para pasar los datos del formulario anterior a una plantilla de word. Por último, se crea una tarea automática para enviar por correo el documento creado anteriormente, y se crea una tarea de confirmación, tras la que el proceso finaliza.

### Diagrama BPMN

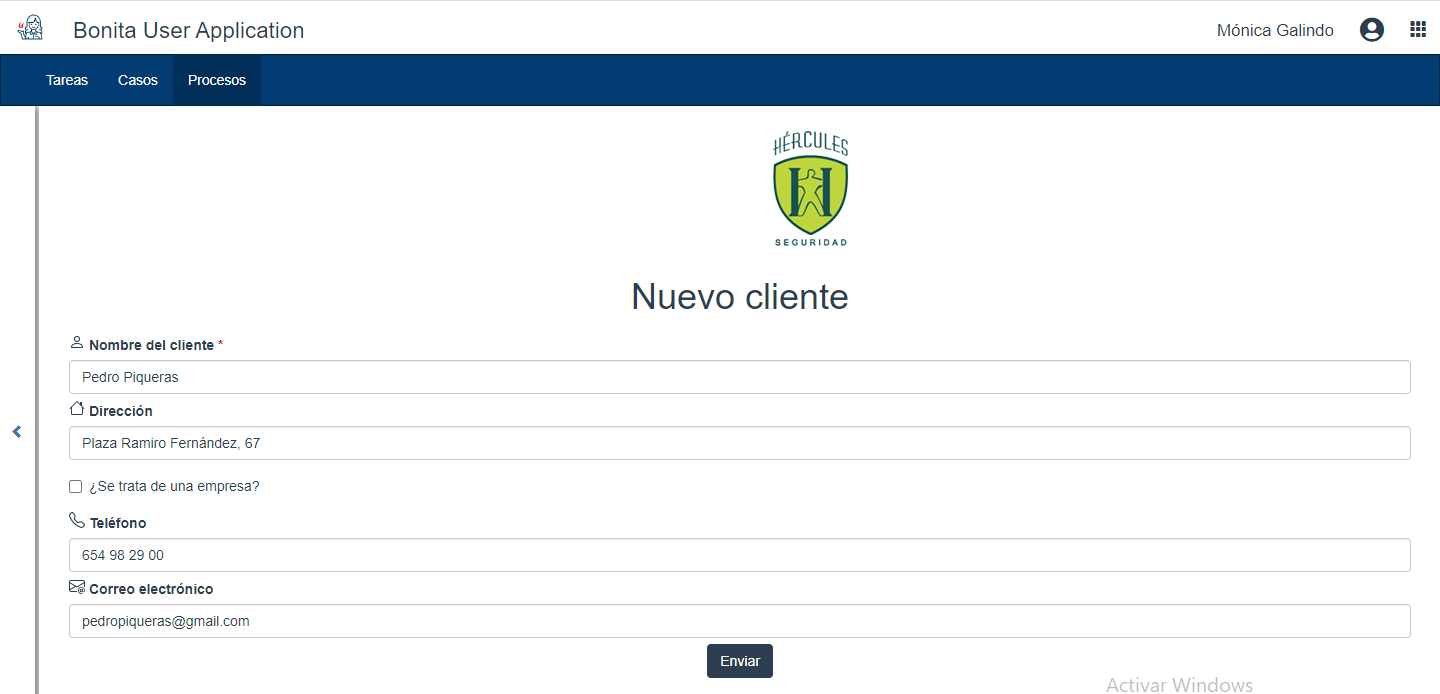


### Resumen de características

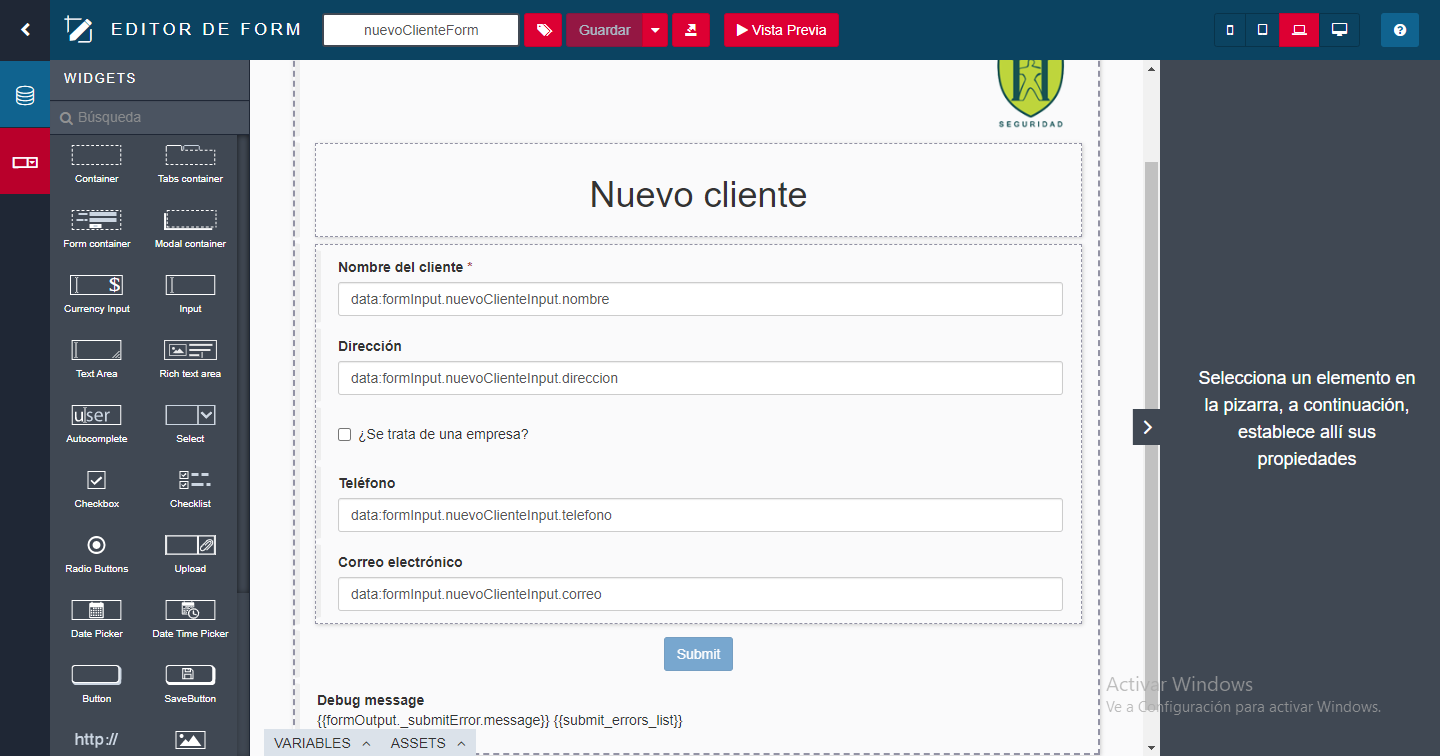
| ¿Hay formulario de instanciación? | Si |
| --- | --- |
| Número de actores | 2 |
| Número de filtros de actor | 1 |
| Número de contratos | 4 |
| Número de variables de negocio | 1 |
| Número de variables de proceso Boolean | 0 |
| Número de variables de proceso Texto | 2 |
| Número de variables de proceso Fecha | 1 |
| Número de variables de proceso Numérica | 3 |
| Número de parámetros | 0 |
| Número de documentos | 2 |
| Número de tareas humanas (con formulario) | 5 |
| Número de tareas de servicio | 2 |
| Número de tareas de script | 0 |
| Número de compuertas XOR | 1 |
| Número de compuertas AND | 0 |
| Número de compuertas OR | 0 |
| Número de eventos temporizadores | 0 |
| Número de otros eventos (distintos de inicio, fin o terminación) | 0 |
| Número de conectores | 2 |

### nuevoClienteForm

* Tarea asociada: nuevoCliente
* Ejecución



* Diseño



* Variables
  + formInput
    - Tipo: JSON
    - Contenido:

{

"nuevoClienteInput" : {

"nombre" : "",

"direccion":"",

"empresa" : false,

"telefono" : "",

"correo" : ""

}

}

* + formOutput
    - Tipo: Javascript
    - Contenido:

return {

nuevoClienteInput: $data.formInput.nuevoClienteInput

}

* Resumen de características del formulario

| Número de widgets diferentes | 5 |
| --- | --- |
| Número de variables JSON | 1 |
| Número de variables String | 0 |
| Número de variables Javascript | 2 |
| Número de variables External API | 0 |
| Número de variables Business data | 0 |
| Número de variables URL Parameter | 0 |
| Número de assets usados | 0 |
| Número de widgets personalizados usados | 0 |

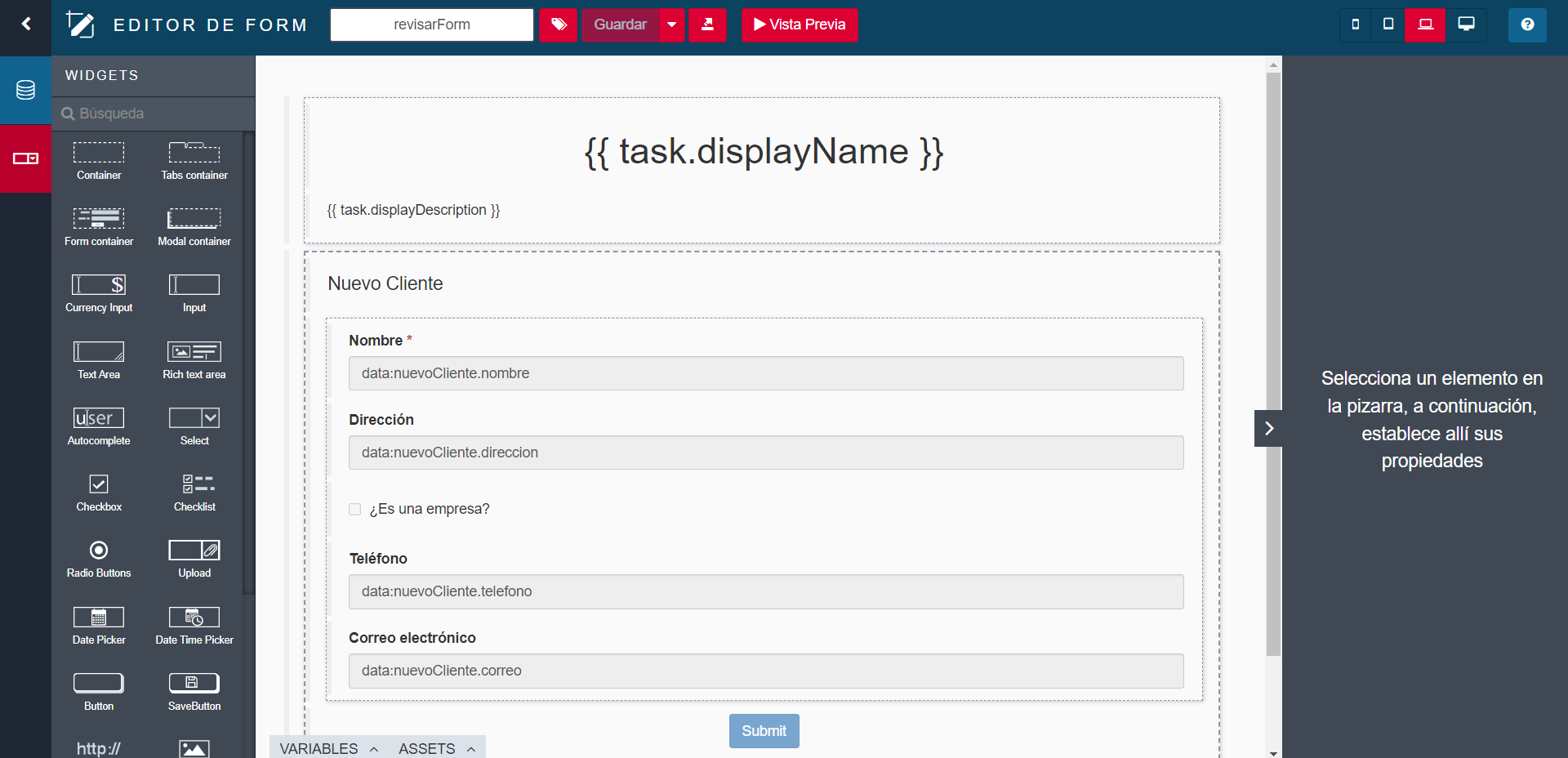
### 

### revisarForm

* Tarea asociada: Revisar cliente
* Ejecución



* Diseño



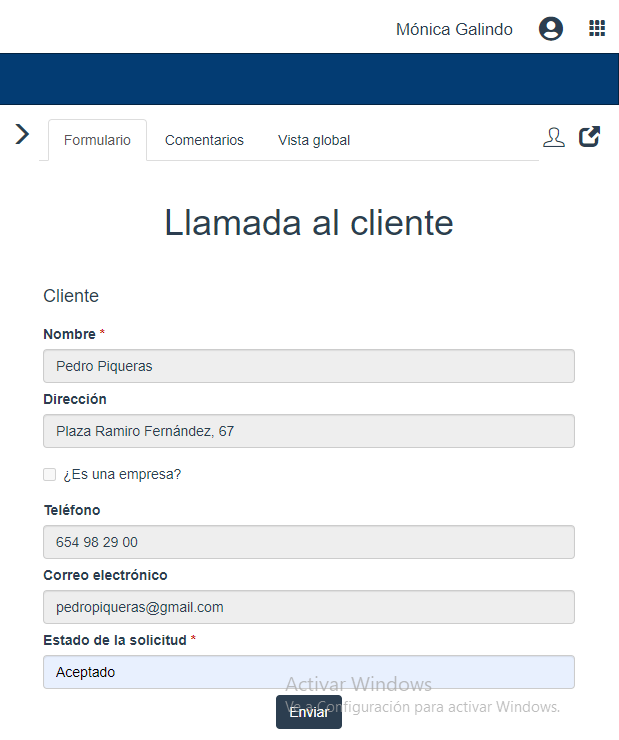
* Variables
  + nuevoCliente
    - Tipo: External API
    - Contenido: ../{{context.nuevoCliente\_ref.link}}
  + nuevoCliente\_sistemas
    - Tipo: External API
    - Contenido: {{nuevoCliente|lazyRef:'sistemas'}}
* Resumen de características del formulario

| Número de widgets diferentes | 6 |
| --- | --- |
| Número de variables JSON | 0 |
| Número de variables String | 0 |
| Número de variables Javascript | 2 |
| Número de variables External API | 2 |
| Número de variables Business data | 0 |
| Número de variables URL Parameter | 0 |
| Número de assets usados | 0 |
| Número de widgets personalizados usados | 0 |

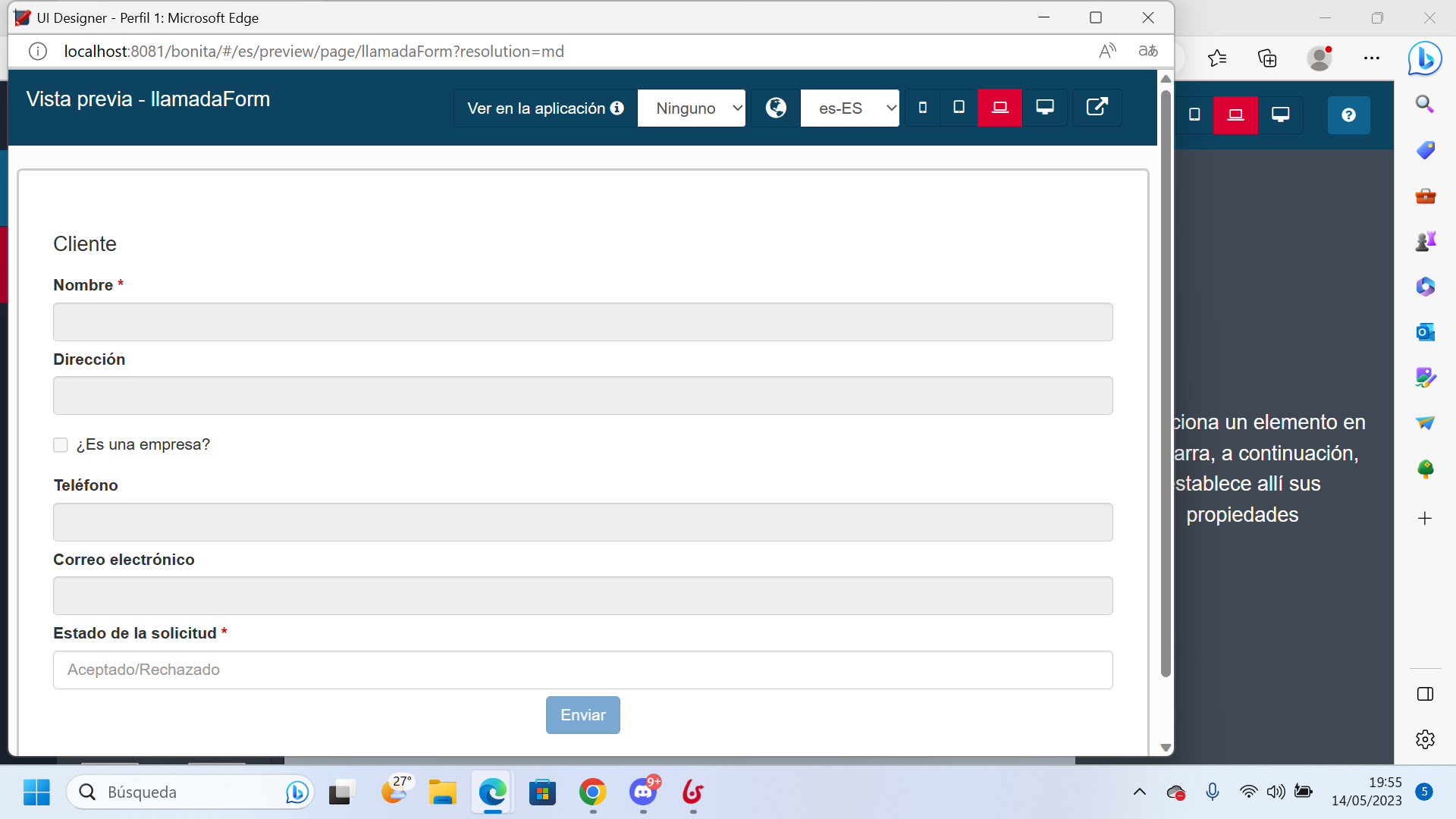
### 

### llamadaForm

* Tarea asociada: Llamada al cliente
* Ejecución



* Diseño



* Variables
  + context
    - Tipo: External API
    - Contenido: ../API/bpm/userTask/{{taskId}}/context
  + formInput
    - Tipo: JSON
    - Contenido: { "estadoInput" : "" }
  + formOutput
    - Tipo: Javascript
    - Contenido: if( $data.nuevoCliente && $data.nuevoCliente\_sistemas ){ //attach lazy references variables to parent variables $data.nuevoCliente.sistemas = $data.nuevoCliente\_sistemas; return { //map nuevoCliente variable to expected task contract input nuevoClienteInput: { nombre: $data.nuevoCliente.nombre !== undefined ? $data.nuevoCliente.nombre : null, direccion: $data.nuevoCliente.direccion !== undefined ? $data.nuevoCliente.direccion : null, empresa: $data.nuevoCliente.empresa !== undefined ? $data.nuevoCliente.empresa : null, telefono: $data.nuevoCliente.telefono !== undefined ? $data.nuevoCliente.telefono : null, correo: $data.nuevoCliente.correo !== undefined ? $data.nuevoCliente.correo : null }, estadoInput: $data.formInput.estadoInput } }
  + nuevoCliente
    - Tipo: External API
    - Contenido: ../{{context.nuevoCliente\_ref.link}}
  + nuevoCliente\_sistemas
    - Tipo: External API
    - Contenido: {{nuevoCliente|lazyRef:'sistemas'}}

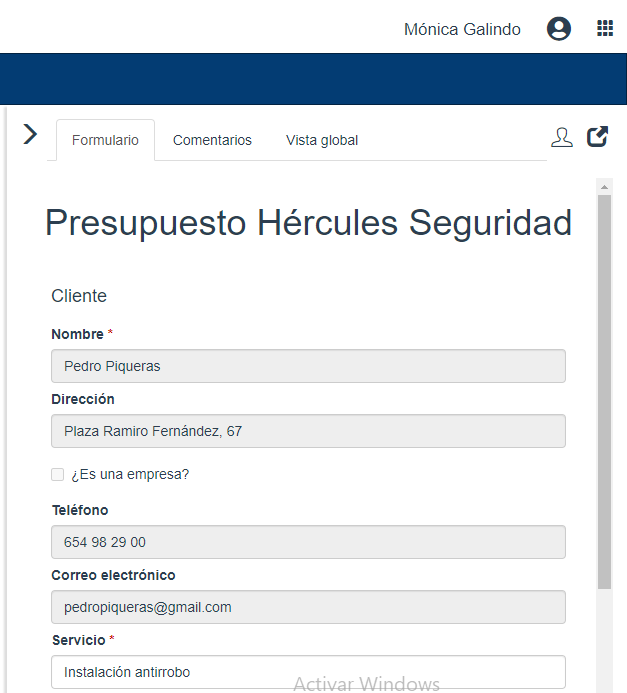
Resumen de características del formulario

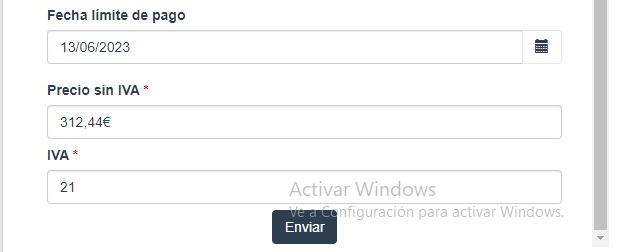
| Número de widgets diferentes | 3 |
| --- | --- |
| Número de variables JSON | 1 |
| Número de variables String |  |
| Número de variables Javascript | 2 |
| Número de variables External API | 4 |
| Número de variables Business data |  |
| Número de variables URL Parameter | 1 |
| Número de assets usados |  |
| Número de widgets personalizados usados |  |

### 

### presupuestoForm

* Tarea asociada: Realizar presupuesto
* Ejecución





* Diseño



* Variables
  + nuevoCliente
    - Tipo: External API
    - Contenido: ../{{context.nuevoCliente\_ref.link}}
  + nuevoCliente\_sistemas
    - Tipo: External API
    - Contenido: {{nuevoCliente|lazyRef:'sistemas'}}

* Resumen de características del formulario

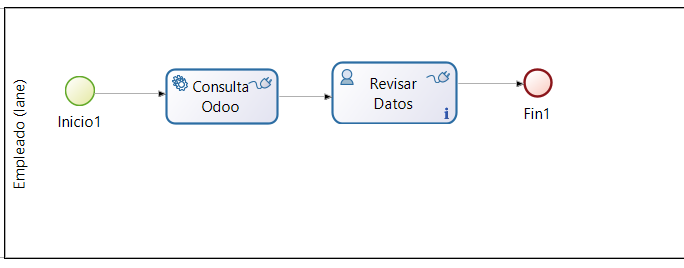
| Número de widgets diferentes | 10 |
| --- | --- |
| Número de variables JSON | 0 |
| Número de variables String | 0 |
| Número de variables Javascript | 2 |
| Número de variables External API | 4 |
| Número de variables Business data | 0 |
| Número de variables URL Parameter | 0 |
| Número de assets usados | 0 |
| Número de widgets personalizados usados | 0 |

## REVISIÓN

El objetivo de este proceso consiste en realizar una revisión de los sistemas que tenemos instalados y que permita conectar con una base de datos externa. Para ello vamos a trabajar con la base de datos de Odoo. Al iniciar el proceso, una tarea de servicio realizará una consulta a la base datos de Odoo en cuanto a los sistemas que tenemos instalados (tabla similar a la del objeto Sistema de este trabajo) Posteriormente mediante una operación dentro de esta tarea se procesarán los datos y se filtrarán, creando una lista que relacione cada tipo de sistema con el número de sistemas de su tipo existentes en Odoo.

Posteriormente en una tarea humana aparecerá un gráfico mostrando dicha proporción calculada, para que el usuario pueda verlo y confirme que todo está correcto. Al final de esta tarea se realizará un INSERT en la base de datos de Odoo dentro de una tabla llamada “revisiones” que almacena el resultado de las revisiones que se han realizado con esta tarea.

### Diagrama BPMN



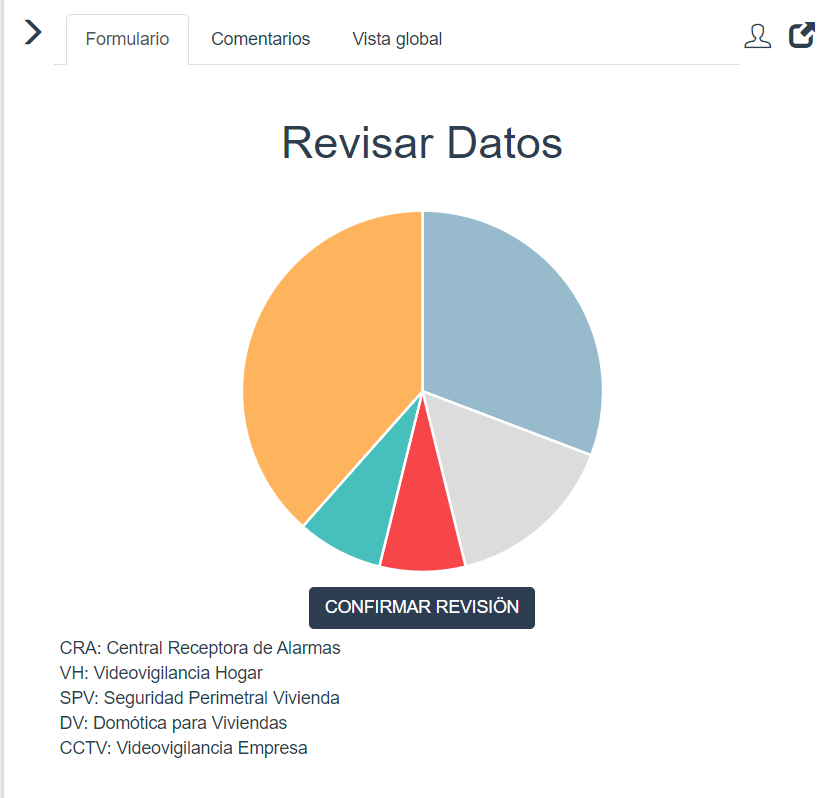
### Resumen de características

NOTA: EXISTEN 3 VARIABLES DE PROCESO DE TIPO java.util.list

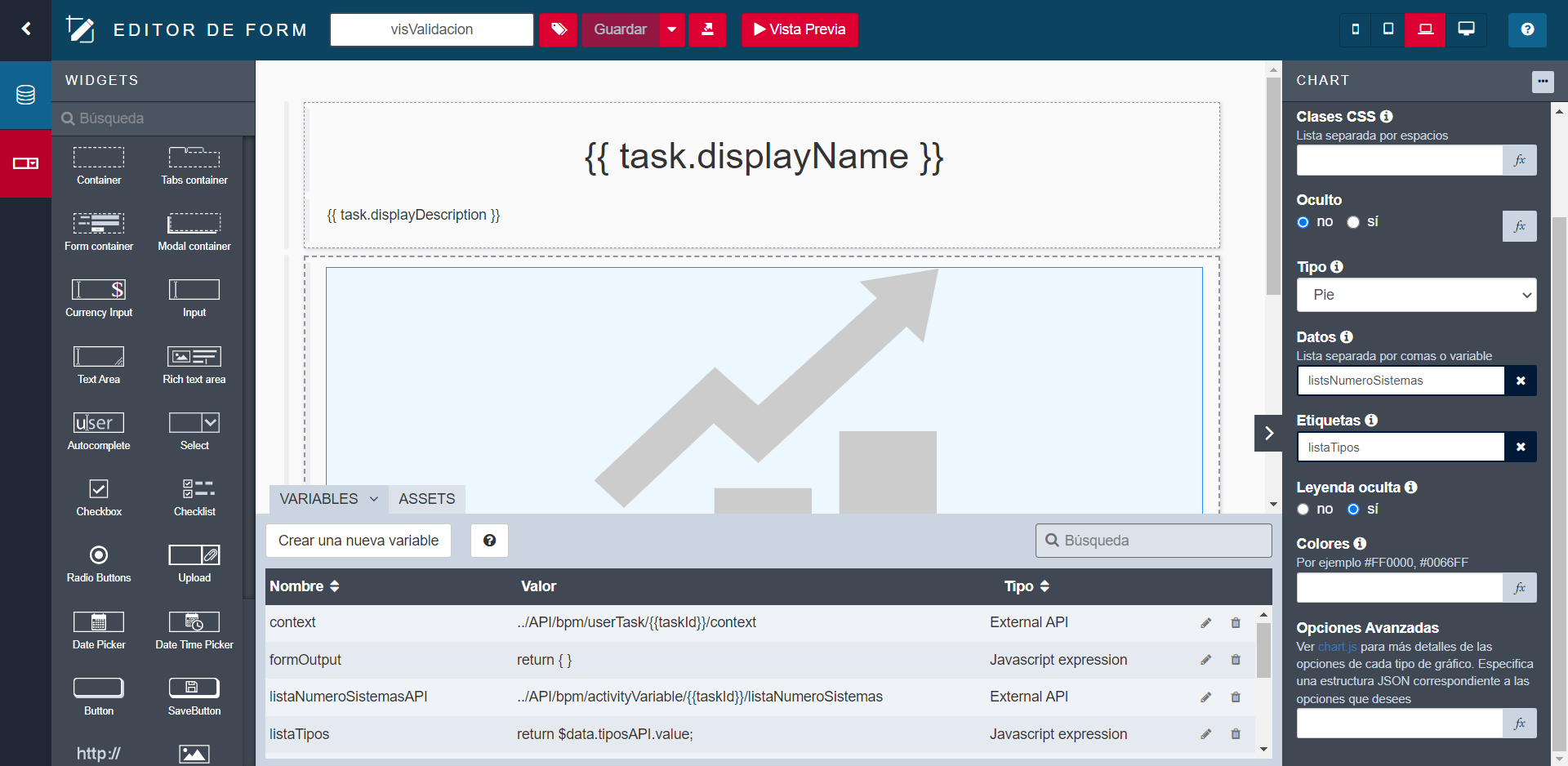
| ¿Hay formulario de instanciación? | No |
| --- | --- |
| Número de actores | 1 |
| Número de filtros de actor | 0 |
| Número de contratos | 0 |
| Número de variables de negocio | 0 |
| Número de variables de proceso Boolean | 0 |
| Número de variables de proceso Texto | 1 |
| Número de variables de proceso Fecha | 0 |
| Número de variables de proceso Numérica | 0 |
| Número de parámetros | 0 |
| Número de documentos | 0 |
| Número de tareas humanas (con formulario) | 1 |
| Número de tareas de servicio | 1 |
| Número de tareas de script | 0 |
| Número de compuertas XOR | 0 |
| Número de compuertas AND | 0 |
| Número de compuertas OR | 0 |
| Número de eventos temporizadores | 0 |
| Número de otros eventos (distintos de inicio, fin o terminación) | 0 |
| Número de conectores | 2 |

### visValidacion

* Tarea asociada: Consulta Odoo
* Ejecución



* Diseño



* Variables
  + listaNumeroSistemasAPI
    - Tipo: External API
    - Contenido: Llamada a la API para conseguir una de las variables de proceso ../API/bpm/activityVariable/{{taskId}}/listaNumeroSistemas
  + listaNumeroSistemas
    - Tipo: Javascript Expression
    - Contenido: Valor resultante de la llamada a la API

return $data.listaNumeroSistemasAPI.value;

* + listaTiposAPI
    - Tipo: External API
    - Contenido: Llamada a la API para conseguir una de las variables de proceso ../API/bpm/activityVariable/{{taskId}}/listaTipos
  + listaTipos
    - Tipo: Javascript Expression
    - Contenido: Valor resultante de la llamada a la API

return $data.listaTipos.value;

* Resumen de características del formulario

| Número de widgets diferentes | 3 |
| --- | --- |
| Número de variables JSON | 0 |
| Número de variables String | 0 |
| Número de variables Javascript | 2 |
| Número de variables External API | 2 |
| Número de variables Business data | 0 |
| Número de variables URL Parameter | 0 |
| Número de assets usados | 0 |
| Número de widgets personalizados usados | 0 |

## CrearSistema

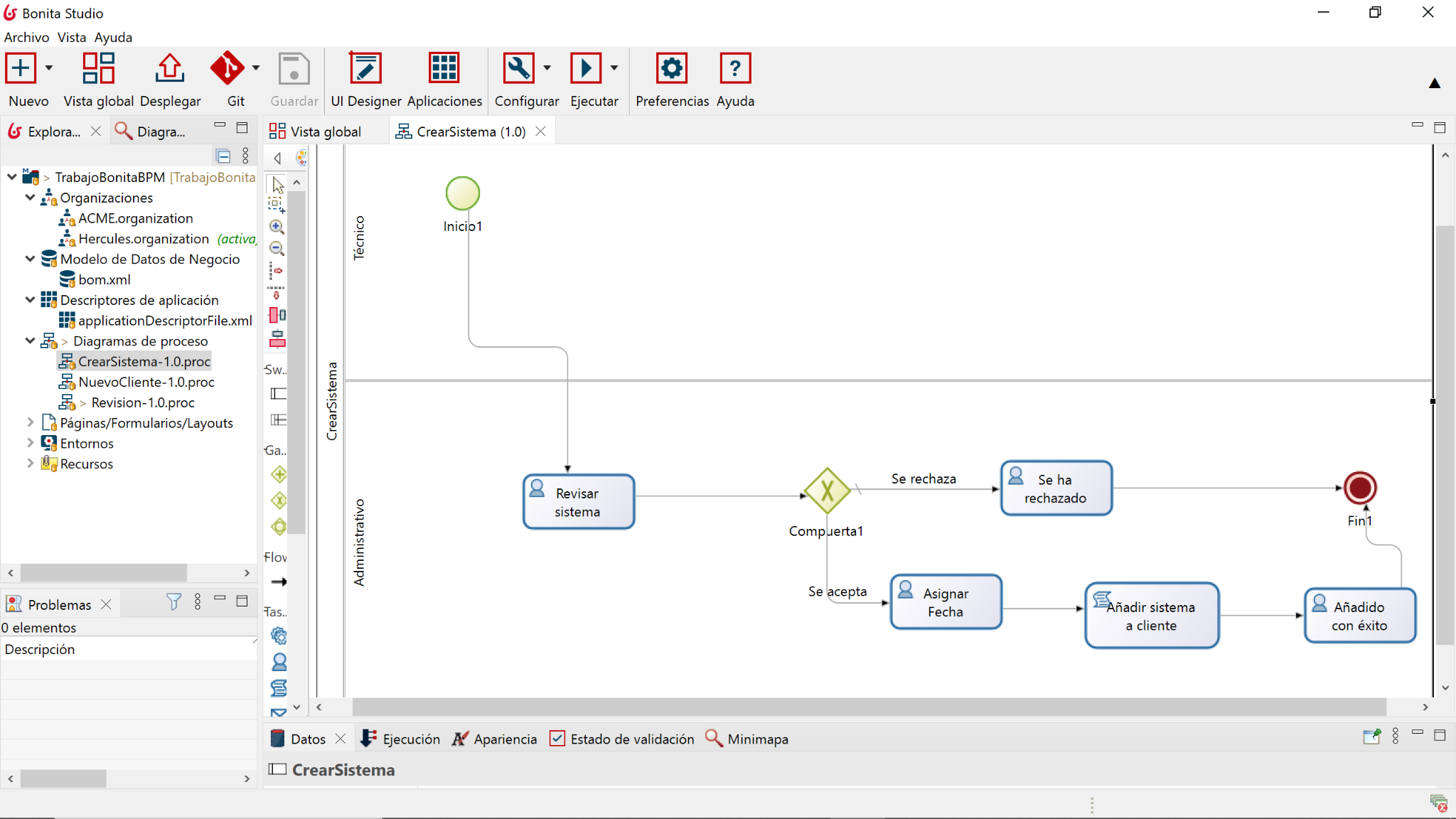
Este proceso se debe ejecutar por un técnico. Además de este, también formará parte del proceso un administrador. Por ello, se han creado dos swimlanes, cada una con un actor definido como técnico y como administrador. Cabe destacar que no se han utilizado filtros de actor porque no es necesario que ambas tareas de administrador las ejecute el mismo usuario.

El proceso comienza siendo ejecutado por un técnico, el cual rellena un formulario de instanciación para crear un sistema (cabe destacar que el campo de “técnico” se rellena automáticamente con el técnico que inicia el proceso y que el campo de “fechaInstalacion” queda sin rellenar de momento).

Una vez se crea este sistema, se genera una tarea para un administrador en la cual se puede aceptar o rechazar. Si se acepta, el campo “aceptado” cambia a “true”.

El valor de dicho campo entra en la compuerta xor. Si es “false”, se crea una tarea indicando que el sistema ha sido rechazado y se finaliza el proceso. Si es “true”, se genera una tarea en la cual el administrador asigna una fecha para que se realice la instalación del sistema, seguida de una tarea de script que añade el sistema al cliente especificado y de una final que indica que el sistema ha sido añadido, después de esto hay otro fin.

### Diagrama BPMN

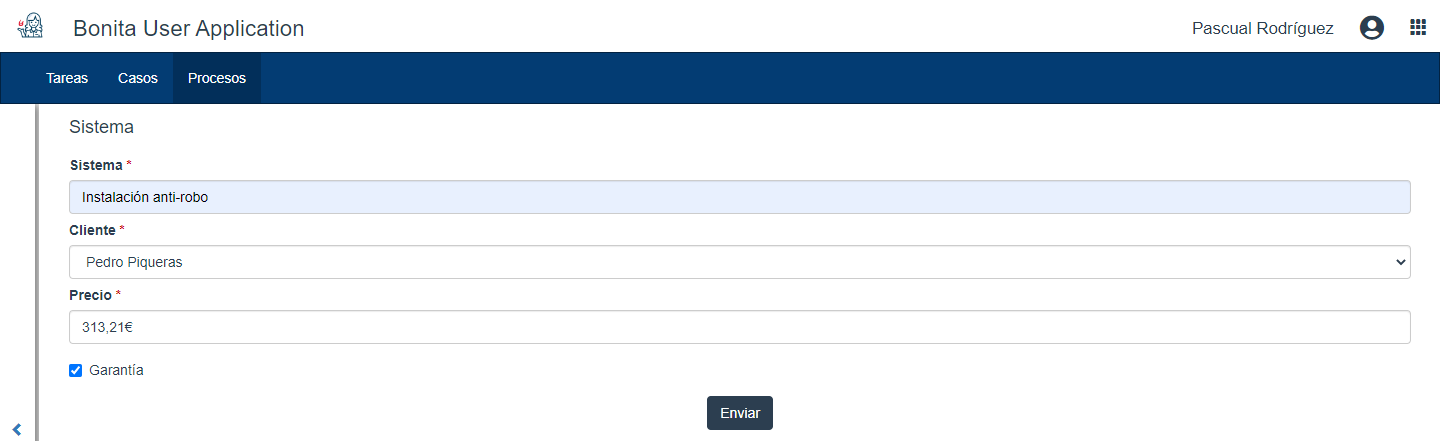


### Resumen de características

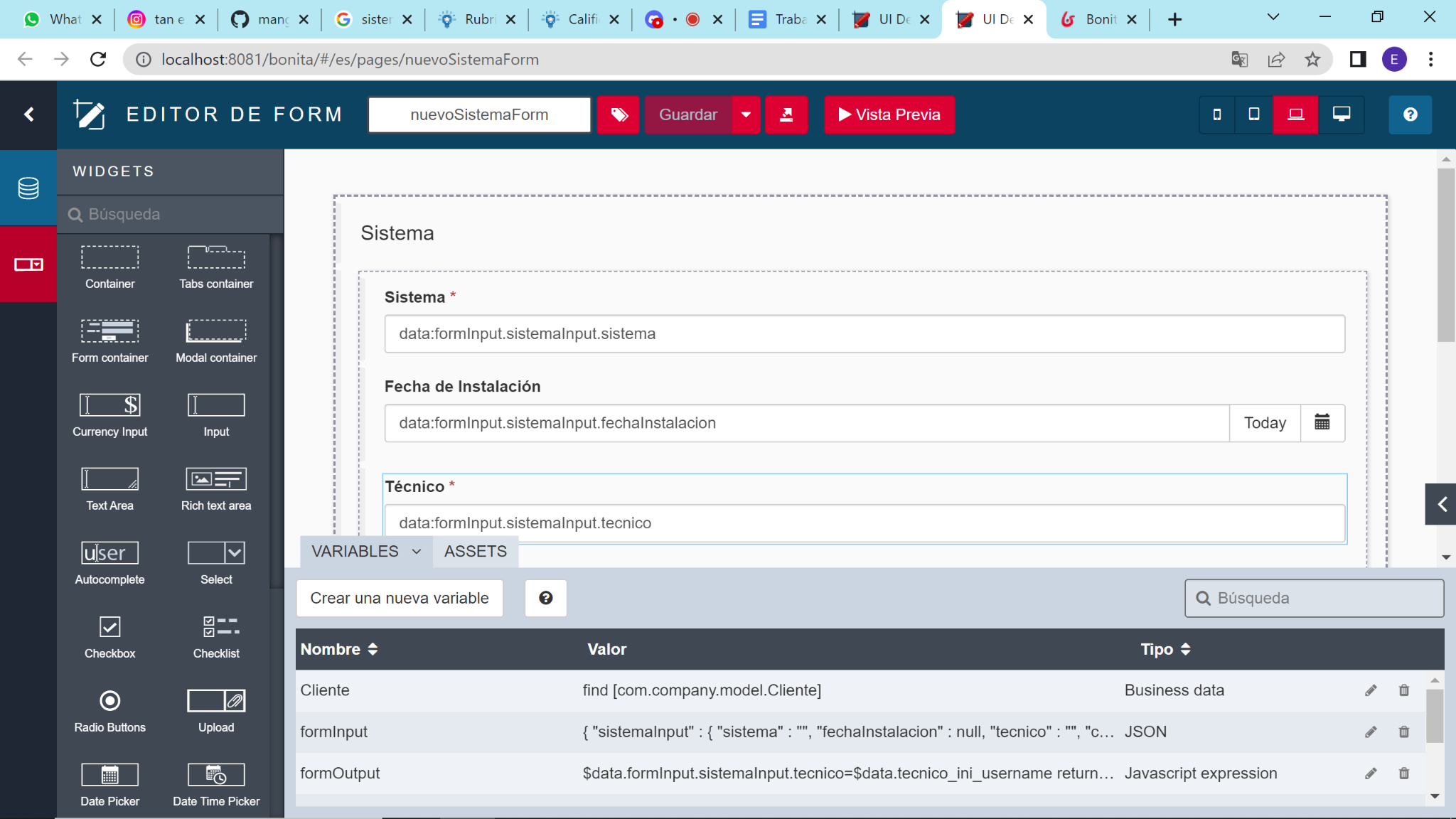
| ¿Hay formulario de instanciación? | Si |
| --- | --- |
| Número de actores | 2 |
| Número de filtros de actor | 0 |
| Número de contratos | 1 |
| Número de variables de negocio | 2 |
| Número de variables de proceso Boolean | 0 |
| Número de variables de proceso Texto | 0 |
| Número de variables de proceso Fecha | 0 |
| Número de variables de proceso Numérica | 0 |
| Número de parámetros | 0 |
| Número de documentos | 0 |
| Número de tareas humanas (con formulario) | 4 |
| Número de tareas de servicio | 0 |
| Número de tareas de script | 1 |
| Número de compuertas XOR | 1 |
| Número de compuertas AND | 0 |
| Número de compuertas OR | 0 |
| Número de eventos temporizadores | 0 |
| Número de otros eventos (distintos de inicio, fin o terminación) | 0 |
| Número de conectores | 0 |

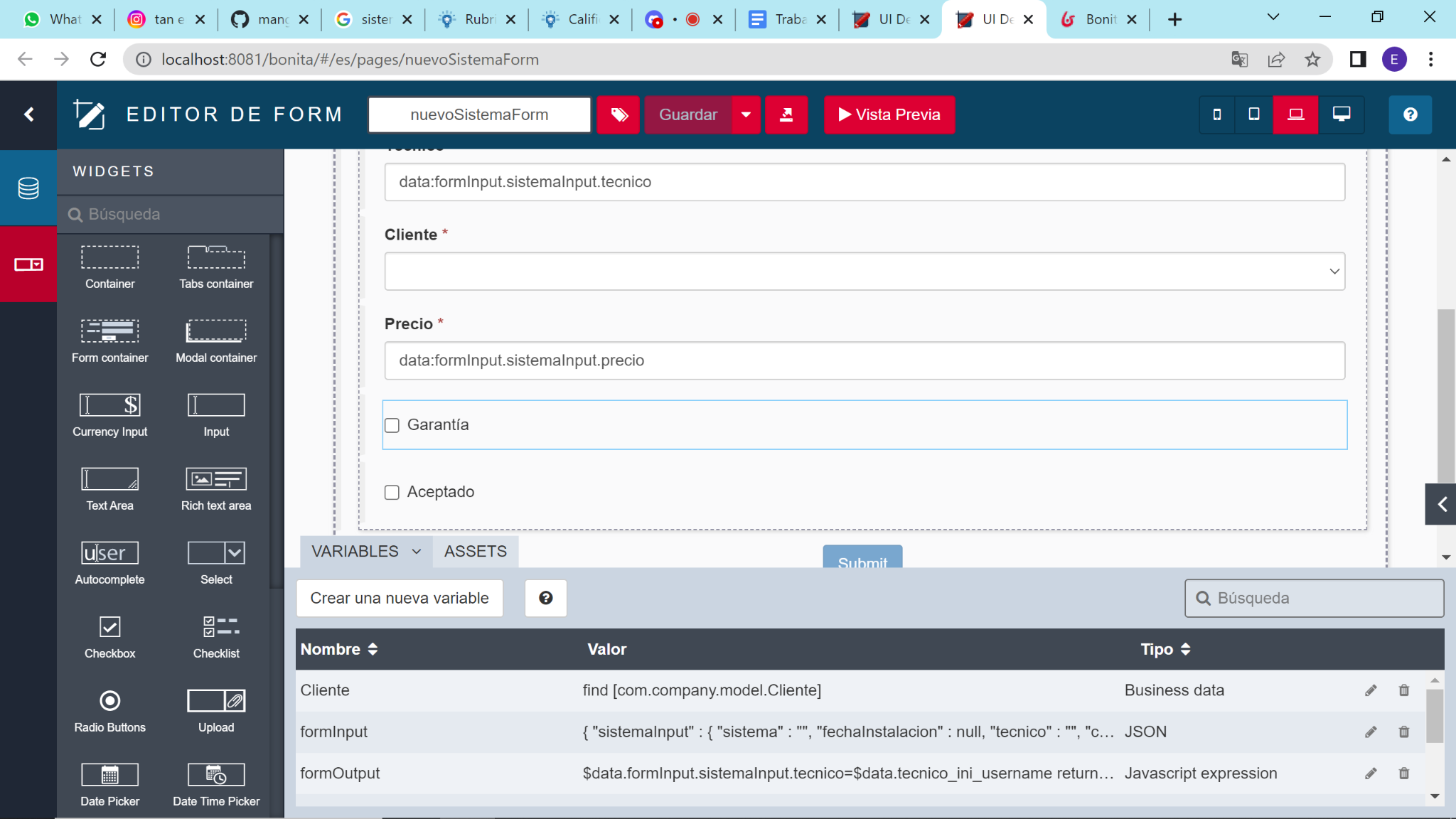
### nuevoSistemaForm

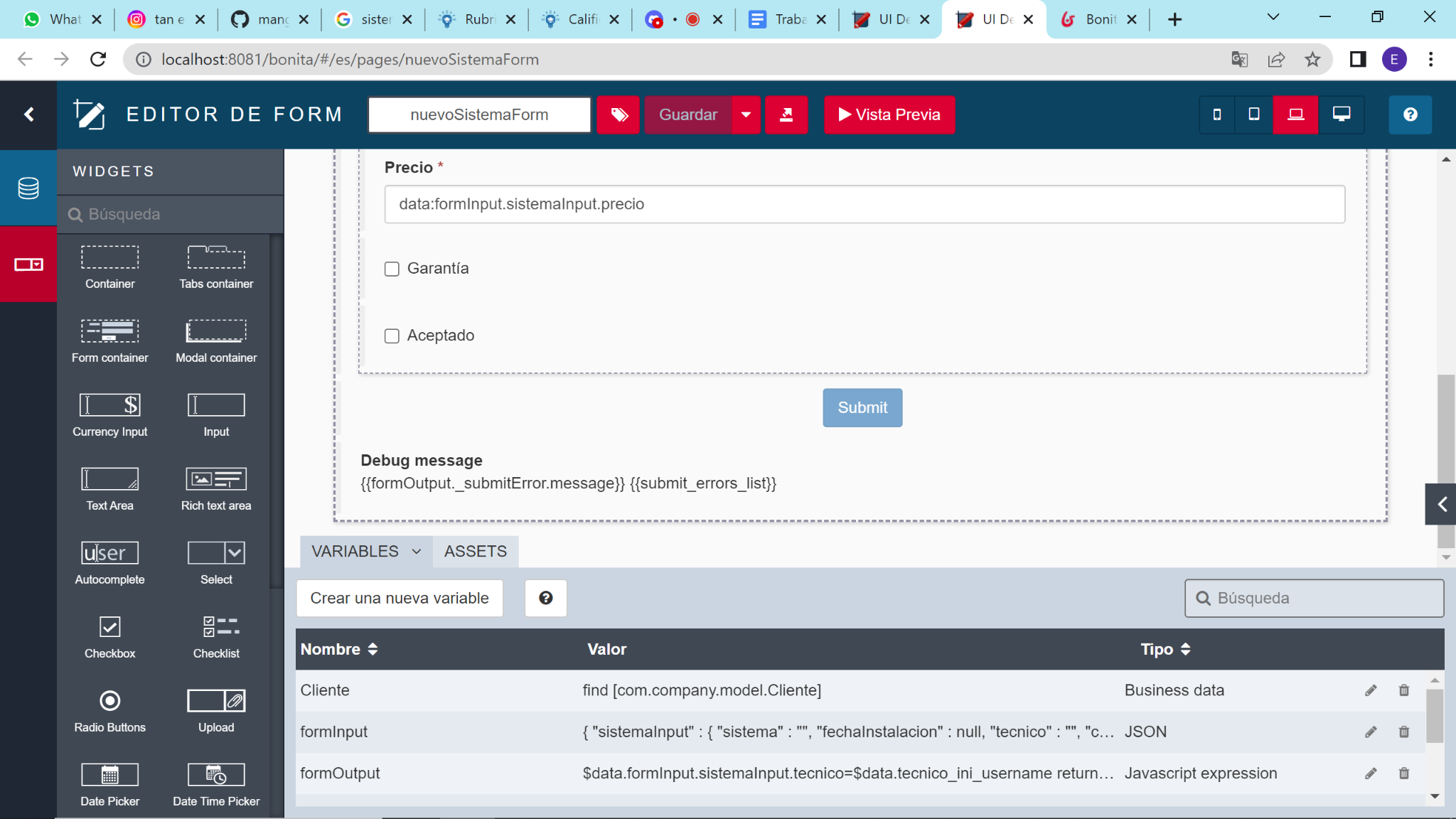
* Tarea asociada: “CrearSistema” (es de instanciación)
* Ejecución



* Diseño







* Variables
  + Cliente
    - Tipo: Business data
    - Contenido: find [com.company.model.Cliente]
  + formInput
    - Tipo: JSON
    - Contenido: {

"sistemaInput" : {

"sistema" : "",

"fechaInstalacion" : null,

"tecnico" : "",

"cliente" : "",

"precio" : 0,

"garantia" : false,

"aceptado" : false

}

}

* + formOutput
    - Tipo: Javascript Expression
    - Contenido:$data.formInput.sistemaInput.tecnico=$data.tecnico\_ini\_username

return {

sistemaInput: $data.formInput.sistemaInput

}

* + submit\_errors\_list
    - Tipo: Javascript expression
    - Contenido: if($data.formOutput && $data.formOutput.\_submitError && $data.formOutput.\_submitError.explanations){

const liElements = $data.formOutput.\_submitError.explanations

.filter(cause => cause !== null)

.map(cause => "<li>" + cause + "</li>")

.join('');

if(liElements){

return "<ul>" + liElements + "</ul>";

}

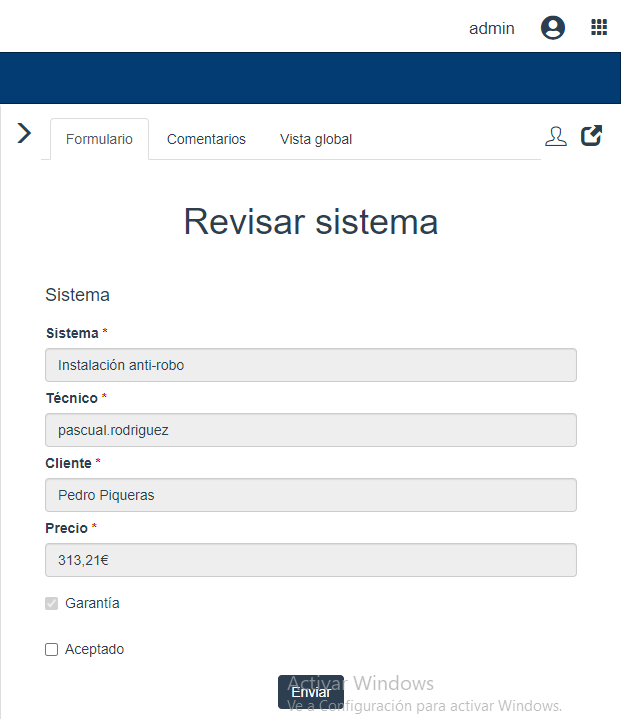
}

* + tecnico\_ini
    - Tipo: External API
    - Contenido: ../API/system/session/1
  + tecnico\_ini\_username
    - Tipo: Javascript expression
    - Contenido: return $data.tecnico\_ini.user\_name;
* Resumen de características del formulario

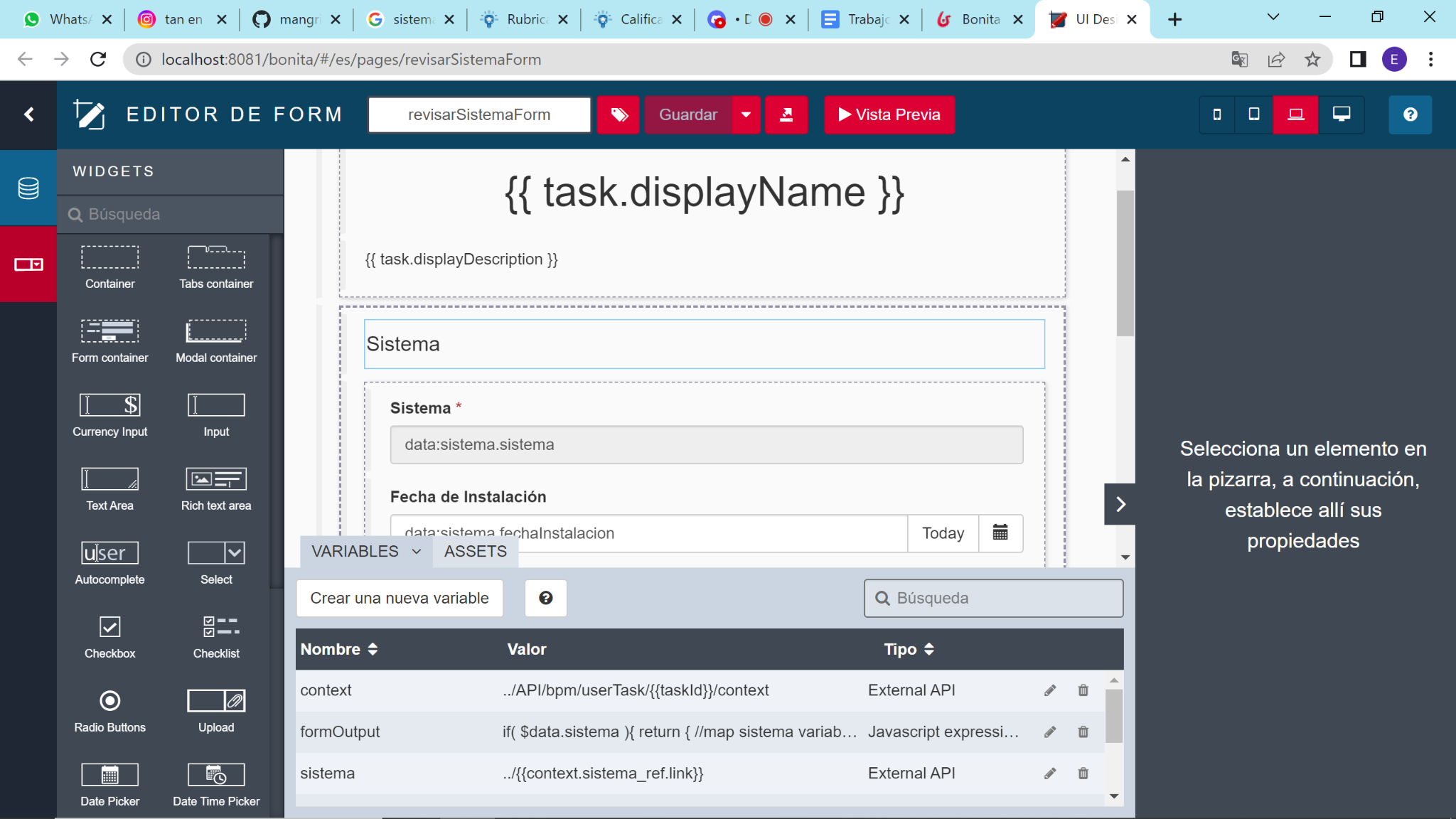
| Número de widgets diferentes | 4 |
| --- | --- |
| Número de variables JSON | 1 |
| Número de variables String | 0 |
| Número de variables Javascript | 3 |
| Número de variables External API | 1 |
| Número de variables Business data | 1 |
| Número de variables URL Parameter | 0 |
| Número de assets usados | 0 |
| Número de widgets personalizados usados | 0 |

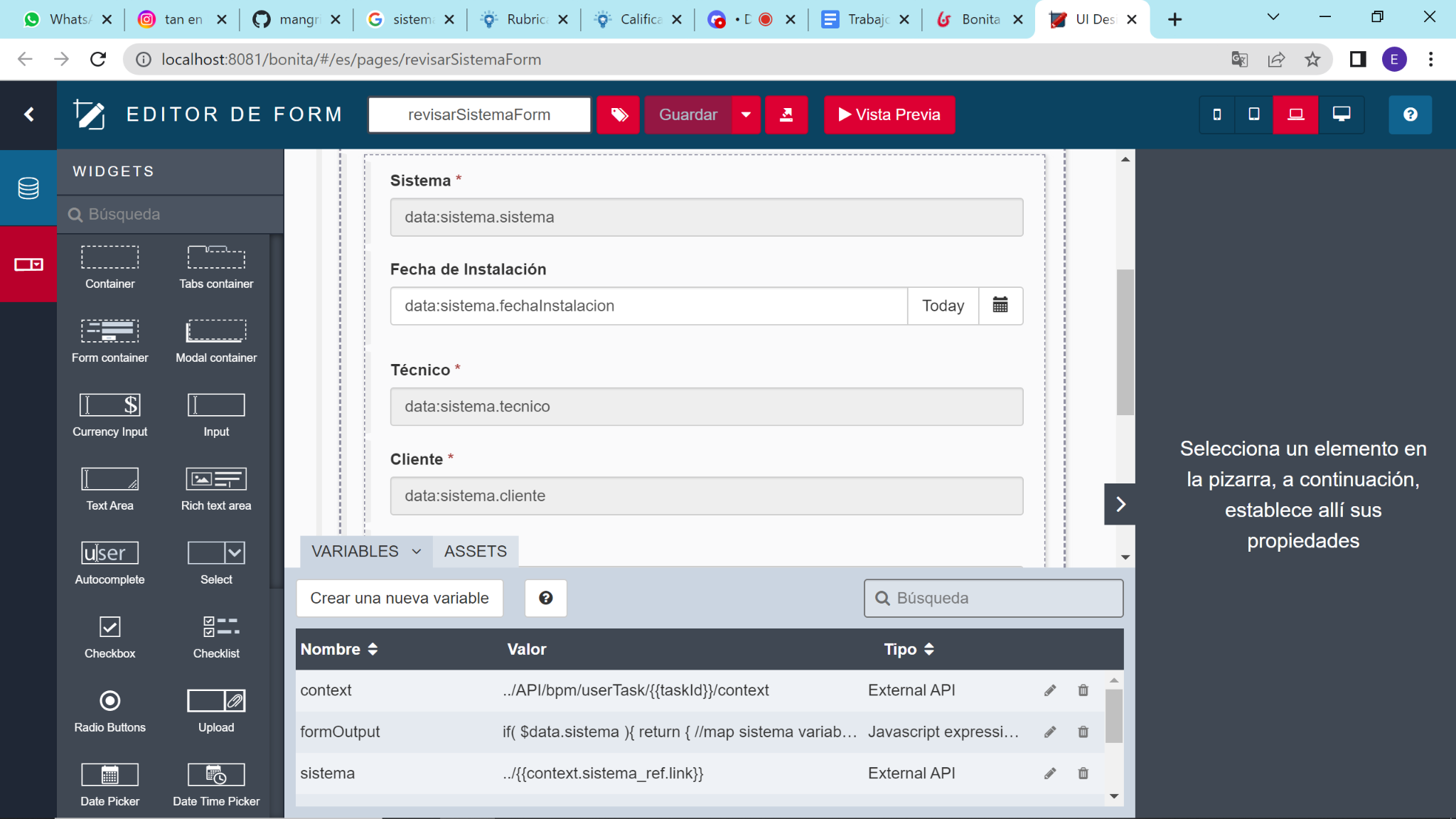
### revisarSistemaForm

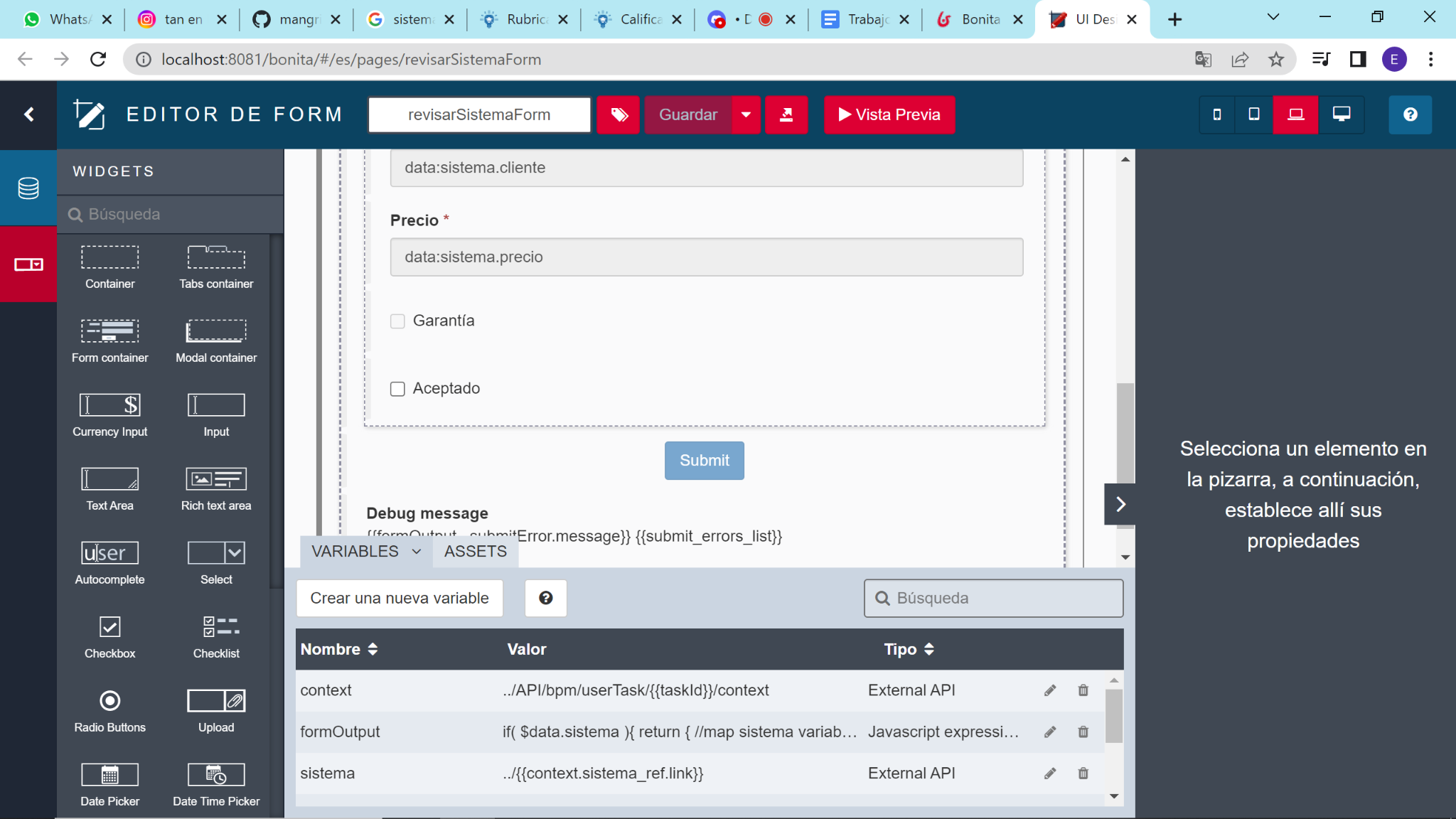
* Tarea asociada: RevisarSistema
* Ejecución



* Diseño







* Variables
  + context
    - Tipo: External API
    - Contenido: ../API/bpm/userTask/{{taskId}}/context
  + sistema
    - Tipo:External API
    - Contenido: ../{{context.sistema\_ref.link}}
  + task
    - Tipo: External API
    - Contenido: ../API/bpm/userTask/{{taskId}}
  + taskId
    - Tipo:URL parameter
    - Contenido: id
* Resumen de características del formulario

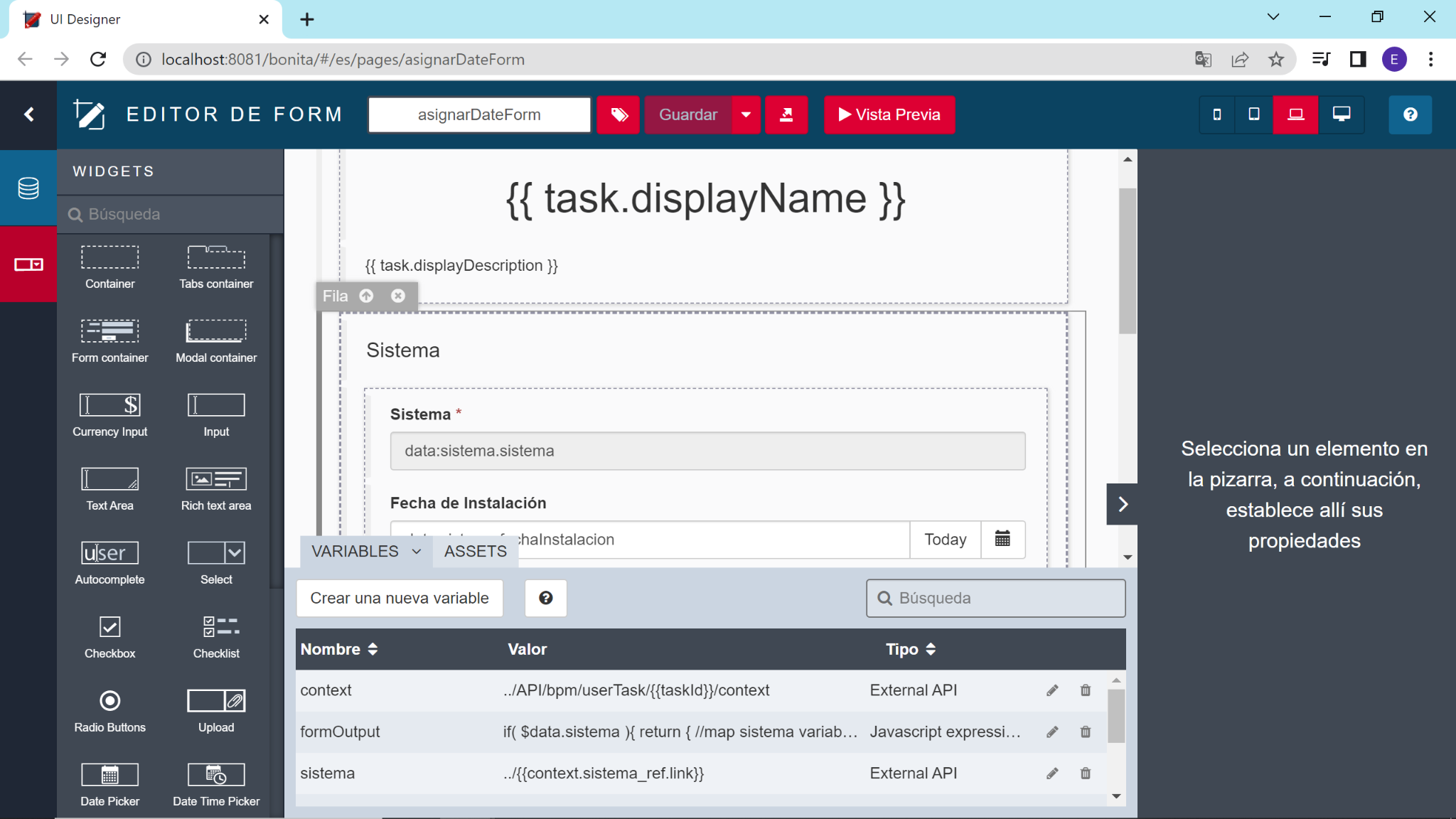
| Número de widgets diferentes | 4 |
| --- | --- |
| Número de variables JSON | 0 |
| Número de variables String | 0 |
| Número de variables Javascript | 2 |
| Número de variables External API | 3 |
| Número de variables Business data | 0 |
| Número de variables URL Parameter | 1 |
| Número de assets usados | 0 |
| Número de widgets personalizados usados | 0 |

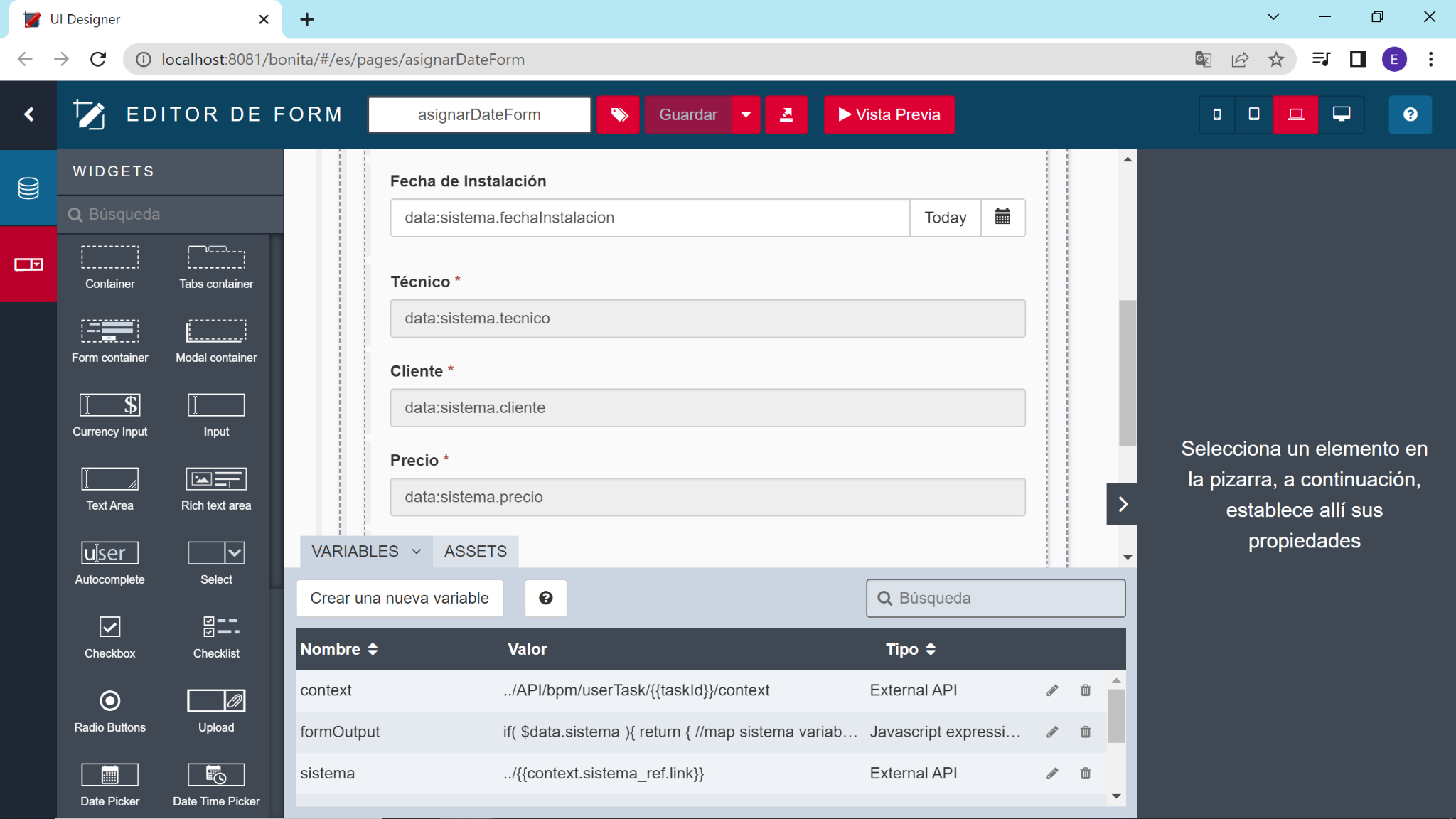
### asignarDateForm

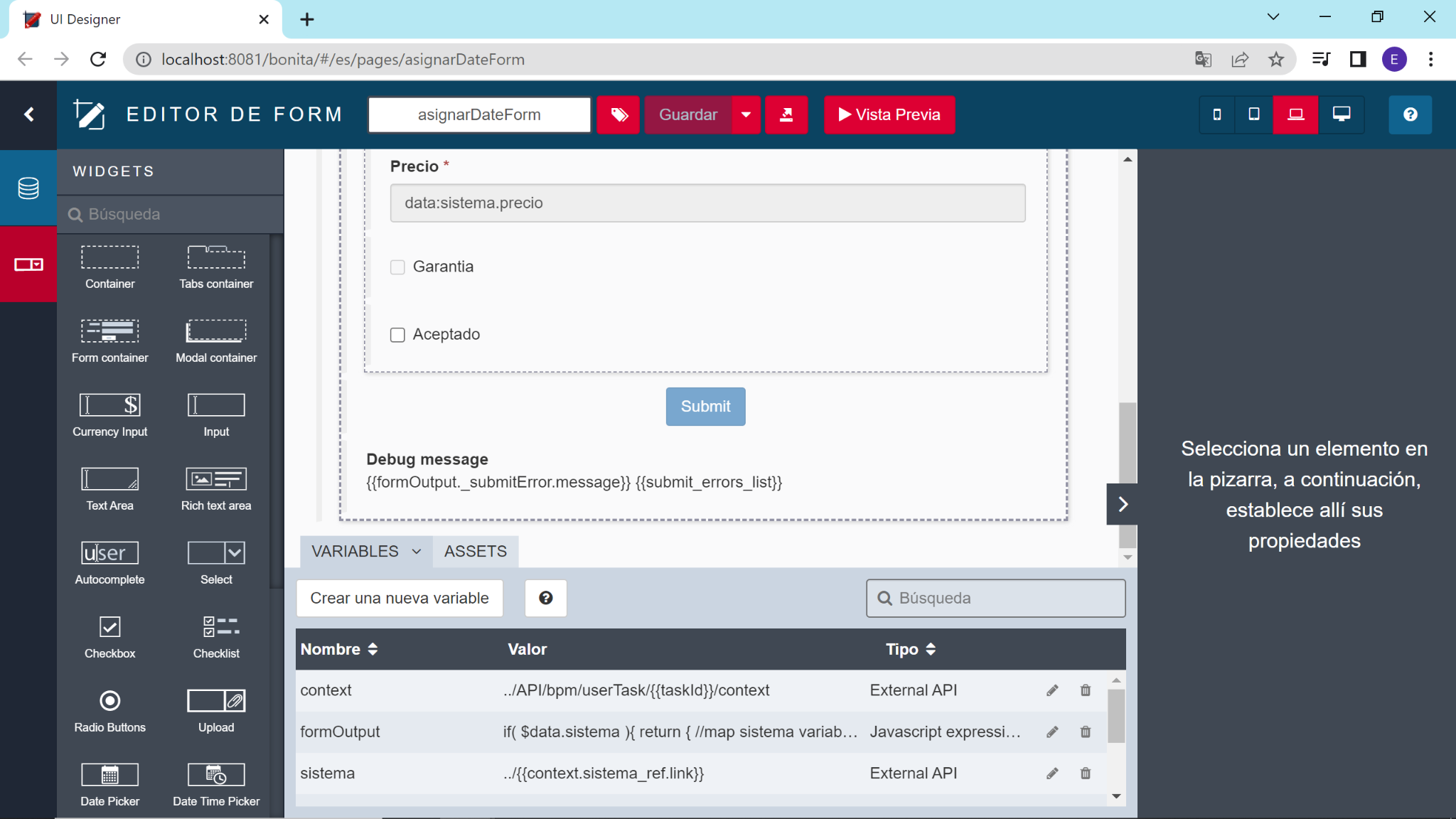
* Tarea asociada: Asignar fecha
* Ejecución



* Diseño







* Variables
  + context
    - Tipo: External API
    - Contenido: ../API/bpm/userTask/{{taskId}}/context
  + sistema
    - Tipo:External API
    - Contenido: ../{{context.sistema\_ref.link}}
  + task
    - Tipo: External API
    - Contenido: ../API/bpm/userTask/{{taskId}}
  + taskId
    - Tipo:URL parameter
    - Contenido: id
* Resumen de características del formulario

| Número de widgets diferentes | 4 |
| --- | --- |
| Número de variables JSON | 0 |
| Número de variables String | 0 |
| Número de variables Javascript | 2 |
| Número de variables External API | 3 |
| Número de variables Business data | 0 |
| Número de variables URL Parameter | 1 |
| Número de assets usados | 0 |
| Número de widgets personalizados usados | 0 |