

**Asesoría especializada para el desarrollo de
herramientas de microsimulación fiscal**

Implementación y arquitectura de los simuladores en el IEF.

jaime.villanueva@ief.minhap.es

- Implementación y arquitectura de los simuladores en el IEF.
Preprocesamiento de los datos de entrada en los simuladores.
- El Impuesto a la Renta de las Personas Físicas en España – el IRPF.
- La interfaz del modelo de simulación del Impuesto sobre la Renta de las Personas Físicas del IEF
- El modelo de simulación del Impuesto sobre la Renta de las Personas Físicas del IEF. Ejercicios prácticos.
- El modelo de simulación del Impuesto de Sociedades del IEF.

1. **Introducción**
2. Tipos de simuladores
 - I. Simuladores de IRPF
 - a) Objetivos.
 - b) Datos de entrada del simulador.
 - c) Ejecución del simulador.
 - d) Resultados
 - II. Simuladores del Impuestos de sociedades.
 - III. Simuladores IVA.
 - IV. Simuladores de pensiones.
3. Arquitectura de simuladores
 - I. Análisis funcional: casos de uso.
 - II. Descomposición en subsistemas.
 - III. Arquitectura lógica.
 - a) Interfaz web del simulador. Entorno de desarrollo.
 - b) Base de datos.
 - c) Servicios Web.
 - d) Módulo servidor. Servidor WF.
 - IV. Arquitectura física.

1. Introducción

- Uno de los **objetivos prioritarios** del **Instituto de Estudios Fiscales (IEF)**, dentro del **ámbito de la investigación**, es la **disponibilidad de útiles para la microsimulación fiscal**.
- En los últimos años, se han **desarrollado herramientas** de **microsimulación** de **impuestos** y de **gasto**.
 - La **idea subyacente** a los microsimuladores es bien **sencilla**: se parte de un **escenario** de agentes contribuyentes o perceptores de prestaciones que **podemos caracterizar a partir de datos desagregados** (microdatos)
 - Si quisiéramos **comparar** la **situación real con** cualquier otro **escenario inventado**, deberíamos definir el nuevo escenario simulado y someter a los **agentes estudiados** a las **nuevas circunstancias** para ver cuál es la **situación alternativa**.
- El interés de las herramientas de **microsimulación fiscal** se **centra** básicamente en el **conocimiento** de la **nueva recaudación**, el **análisis** de **ganadores y perdedores**, y el **análisis distributivo y** de **bienestar** (intereses generales de la actuación del sector público).

1. Introducción
2. Tipos de simuladores
 - I. Simuladores de IRPF
 - a) Objetivos.
 - b) Datos de entrada del simulador.
 - c) Ejecución del simulador.
 - d) Resultados
 - II. Simuladores del Impuestos de sociedades.
 - III. Simuladores IVA.
 - IV. Simuladores de pensiones.
3. Arquitectura de simuladores
 - I. Análisis funcional: casos de uso.
 - II. Descomposición en subsistemas.
 - III. Arquitectura lógica.
 - a) Interfaz web del simulador. Entorno de desarrollo.
 - b) Base de datos.
 - c) Servicios Web.
 - d) Módulo servidor. Servidor WF.
 - IV. Arquitectura física.

2. Tipos de Simuladores

Tipo	Ejemplos
Imposición directa	<ul style="list-style-type: none">- Simuladores de IRPF- Simuladores de IS para PYMES
Imposición indirecta	<ul style="list-style-type: none">- Simulador de IVA e IIEE
TAX-BENEFIT (Imp. y Prest.)	<ul style="list-style-type: none">- Simulador EUROMOD (Unión Europea)- Simulador TAX-FIT (nacional)
Otros	<ul style="list-style-type: none">- Simulador Pensiones

2.1 Simulador IRPF

- El simulador del IRPF del IEF permite realizar un *análisis de recaudación, de efectos distributivos y de ganadores y perdedores, ante la simulación de escenarios de gravamen alternativos*.
- Los **escenarios** analizados serán lo **más flexibles** posibles. Posibilidad de modificación de todos los parámetros que configuran impuesto.
- El **objetivo** es *volver a calcular las liquidaciones del impuesto a un conjunto de declarantes* de los que tenemos información pero *bajo un escenario distinto* al que tenían cuando ellos lo hicieron, ya que se van a introducir **cambios en el impuesto** que van a conducir a *liquidaciones* potencialmente **diferentes**.

2.1 Simulador IRPF. Datos de entrada

Datos de entrada:

- Datos **fiscales**: Muestra de declarantes **IRPF IEF-AEAT**. Formato: **ASCII**.
 - Muestra a partir de las declaraciones del **modelo 100**, cerca de **2.000.000 declaraciones por año** (conjuntas e individuales) mediante **muestreo estratificado aleatorio** que garantiza la **representatividad** de los distintos tipos de renta. **La información de la base de datos no es modificable**.
 - Se puede **crear nuevas variables** durante el proceso de presimulación o preprocesamiento **para corregir errores** y **calcular variables** necesarias. (MJ)
- **Parámetros** para la **simulación**. Permiten simular los distintos escenarios.
 - El investigador puede cambiar los parámetros (alrededor de **1.000** distribuido entre todos los conceptos de renta) a través de una **aplicación web** donde los distintos parámetros van apareciendo en forma de menús.
 - **Tipo de modificaciones**: 1) Modificar la definición de estos conceptos; 2) Su incorporación a las bases; 3) Las reducciones aplicables; 4) Modificar las tarifas; 5) deducciones, etc.
 - **Ejemplo**: variar una cuantía (por ejemplo, cambiar los 3.400 € de reducción por tributación conjunta en el caso de matrimonio, a 0 €, lo que supondría su eliminación).

2.1 Simulador IRPF. Parámetros (1)

Parámetros generales

- Determinación del margen para calificación de ganadores e indiferentes.
- Opción por actualización de población y rentas

Rendimientos

- Trabajo.
- Capital mobiliario.
- Capital inmobiliario.
- Actividades económicas.
- Imputaciones de rentas.
- Ganancias y pérdidas patrimoniales.

Reducciones

- Por tributación conjunta
- Mínimos personales y familiares.
*Mínimo del contribuyente *Mínimo por descendientes *Mínimo por ascendientes *Mínimo por discapacitados
- Por situaciones de dependencia y envejecimiento
*sistema de previsión social. Régimen general *sistema de previsión social. Régimen de aportaciones cónyuge *sistema de previsión social. Régimen discapacitados *sistema de previsión social. Régimen deportistas profesionales *Parámetros adicionales (para reparto de la parte estatal)

2.1 Simulador IRPF. Parámetros (2)

Otras

- Pensiones compensatorias.
- Anualidades por alimentos.
- Parámetros adicionales.
- Reducción por cuotas afiliación.

Deducciones

- Por inversión en vivienda habitual.
- Deducciones generales normativa estatal.
- Deducciones autonómicas (despliegue para cada CC.AA)
- Deducciones que pueden hacer la cuota negativa (por maternidad)

Tarifas

- Estatales
 - *Parte general *Parte especial
- Autonómicas
- Estatales no residentes.
 - *Parte general *Parte especial
- Autonómicas no residentes.
 - *Parte general *Parte especial

¿En qué consiste la ejecución?

- Todos los **datos de entrada** se han **convertido** a **formato propio del entorno de programación**, tanto **la base de datos con información de las declaraciones** como **los parámetros**.
- Durante la ejecución de la simulación:
 - **Disminución carga computacional**: se utilizan los subconjuntos de datos necesarios en cada cálculo.
 - **Recalcula las liquidaciones** una por una teniendo en cuenta la información de los parámetros.
 - Imputa y crea valores cuando así se requiere para poder llegar a la liquidación final.
 - En ocasiones se **desconocen** determinadas **situaciones sociodemográficas** que dan derecho a reducciones, deducciones, etc.

2.1 Simulador IRPF. Resultados (1)

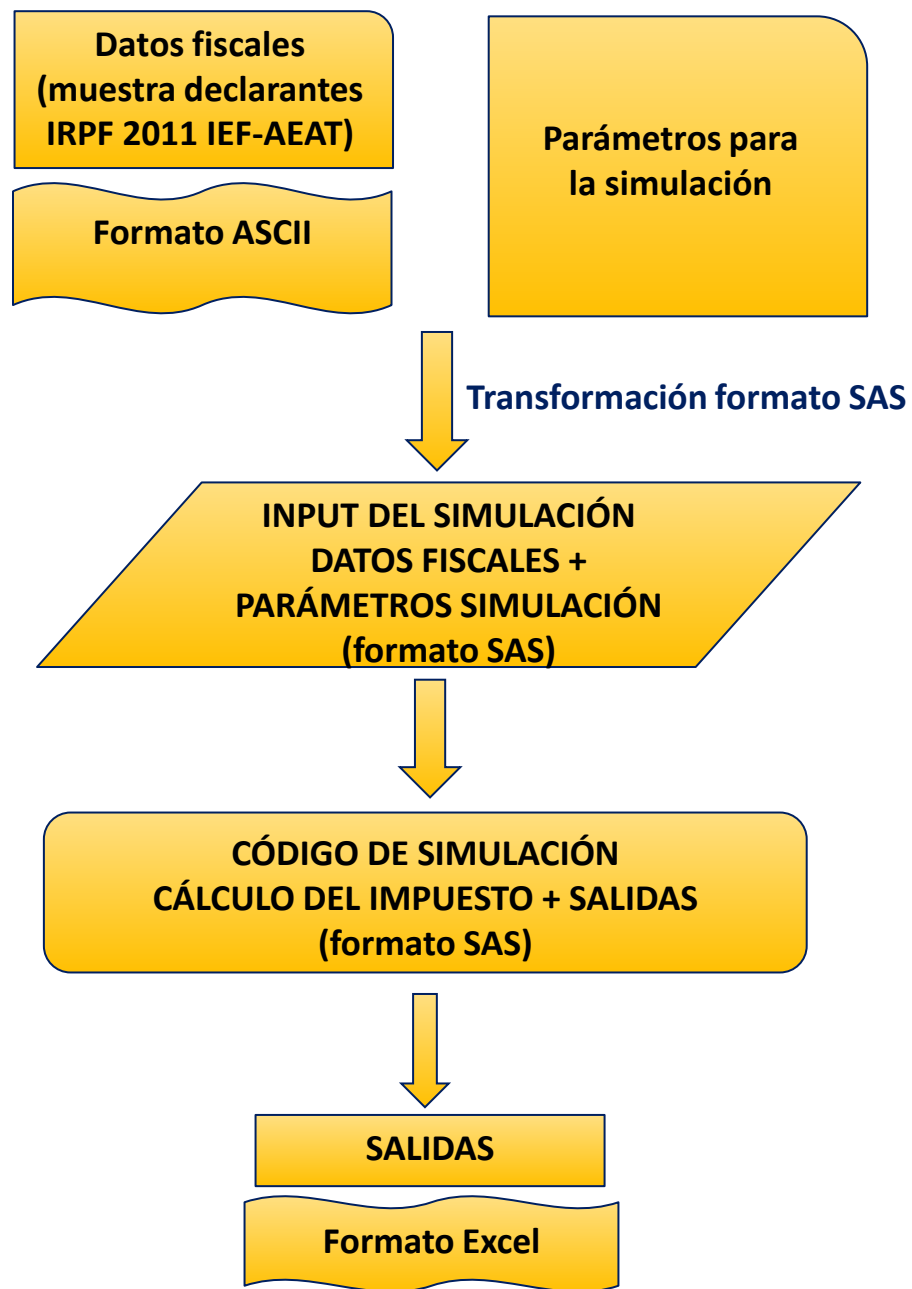
¿Qué resultados se obtienen?:

- Tras hacer la **liquidación** se procede a:
 - **Cálculo de agregados**
 - **Recaudación final**, tanto del escenario base como de la simulación (incremento / decremento recaudación)
 - **Ganadores y perdedores por CCAA**, tipos marginales, decilas de renta, etc.
 - **Índices de desigualdad, progresividad y redistribución**
 - **Gini, Kakwani, Reynolds-Smolensky**
- Los **resultados** se **exportan** a una **tabla Excel**.

2.1 Simulador IRPF. Resultados (2)

Resultados	Descripción	
Agregados	- Información sobre el nº de declaraciones utilizadas, así como las siguientes variables referidas a los datos de la AEAT, la situación simulada y la situación de referencia, computando las diferencias entre las mismas.	<ul style="list-style-type: none"> – Renta antes de impuestos. – Cuota resultante de la autoliquidación. – Renta después de impuestos. – Base liquidable. – Reducciones. – Resto de reducciones (deducciones).
Tipos marginales	<ul style="list-style-type: none"> - Información sobre la cuota resultante de la autoliquidación CRA clasificada por tipos marginales de la base general y la base especial. - Incluye la población que se concentra en cada tramo, el total de CRA antes y después de la reforma y la media de CRA antes y después de la reforma. 	
Índices	En este apartado se calculan los siguientes índices.	<ul style="list-style-type: none"> – Í. de Gini Renta Antes de Impuestos (RAI). – Í. Gini Renta Después de Impuestos (RDI). – Í. de Concentración RDI ordenado por RAI. – Í. de Concentración CRA ordenado por RAI. – Í de Reynolds y Smolensky. – Í. de Kakwani.
Decilas	La información se construye ordenando a los declarantes por su renta antes de impuestos. Para cada decila se ofrece con carácter descriptivo el valor máximo, mínimo, medio y desviación típica, de la renta antes de impuestos, así como el valor total acumulado en la decila de la renta antes de impuestos y la cuota resultante en la situación simulada. En cada decila se calcula el nº total de ganadores, perdedores e indiferentes con la reforma, así como el valor total y medio de la cantidad ganada o perdida.	
Ganadores y perdedores por tarifa	La información referente a ganadores, perdedores e indiferentes, se replica presentándose en grupos determinados por los tramos de la tarifa tanto de la base general como especial	
CC.AA	Para cada una de las CCAA se aporta información sobre la población, el total de renta antes y después de impuestos, la base liquidable, reducciones, deducciones y cuota. También se detalla información sobre el nº total de ganadores, perdedores e indiferentes en cada CA así como de la cantidad media ganada o perdida. El análisis de ganadores y perdedores por decilas se replica a nivel de CCAA de la misma forma que se realiza para el total.	
Tipos de renta	Análisis de ganadores o perdedores por tipos de renta (rentas del trabajo, del capital, de actividades empresariales, o resto de rentas), atendiendo en este caso a la fuente principal de rentas. Para determinar a qué grupo pertenece cada hogar, se toma el criterio de que más de la mitad de las rentas totales proceda de cada una de las fuentes consideradas. Además por cada tipo de rentas, se	

2.1 Simulador IRPF



2.II Impuesto de sociedades

- El simulador del Impuesto de Sociedades para **PYMES** del IEF **permite** realizar un **análisis de recaudación y de ganadores y perdedores** para las **empresas** incluidas en el **Régimen Especial de Reducida Dimensión**
 - **Simulación de escenarios de gravamen alternativos.**
- Diseñado para que los **escenarios** analizados sean lo más **flexibles** posibles,
 - Los parámetros que configuran el impuesto son modificables en su mayoría.

2.III Simulador IVA

- Permite **analizar** los **efectos** de las **reformas** de la **imposición sobre el consumo de las familias**, en concreto, mediante el análisis de reformas **en**:
 - **IVA.**
 - **Impuestos Especiales.**
- Los simuladores permiten **alterar los tipos impositivos** de cualquiera de estos impuestos.

2.IV Simulador de pensiones

- El **simulador de pensiones** presenta como características fundamentales ***el uso de la Encuesta de Condiciones de Vida***, y mediante **ecuaciones de comportamiento** permite **proyectar variables demográficas y de mercado laboral** para toda la ***muestra inicial hasta 2050***.

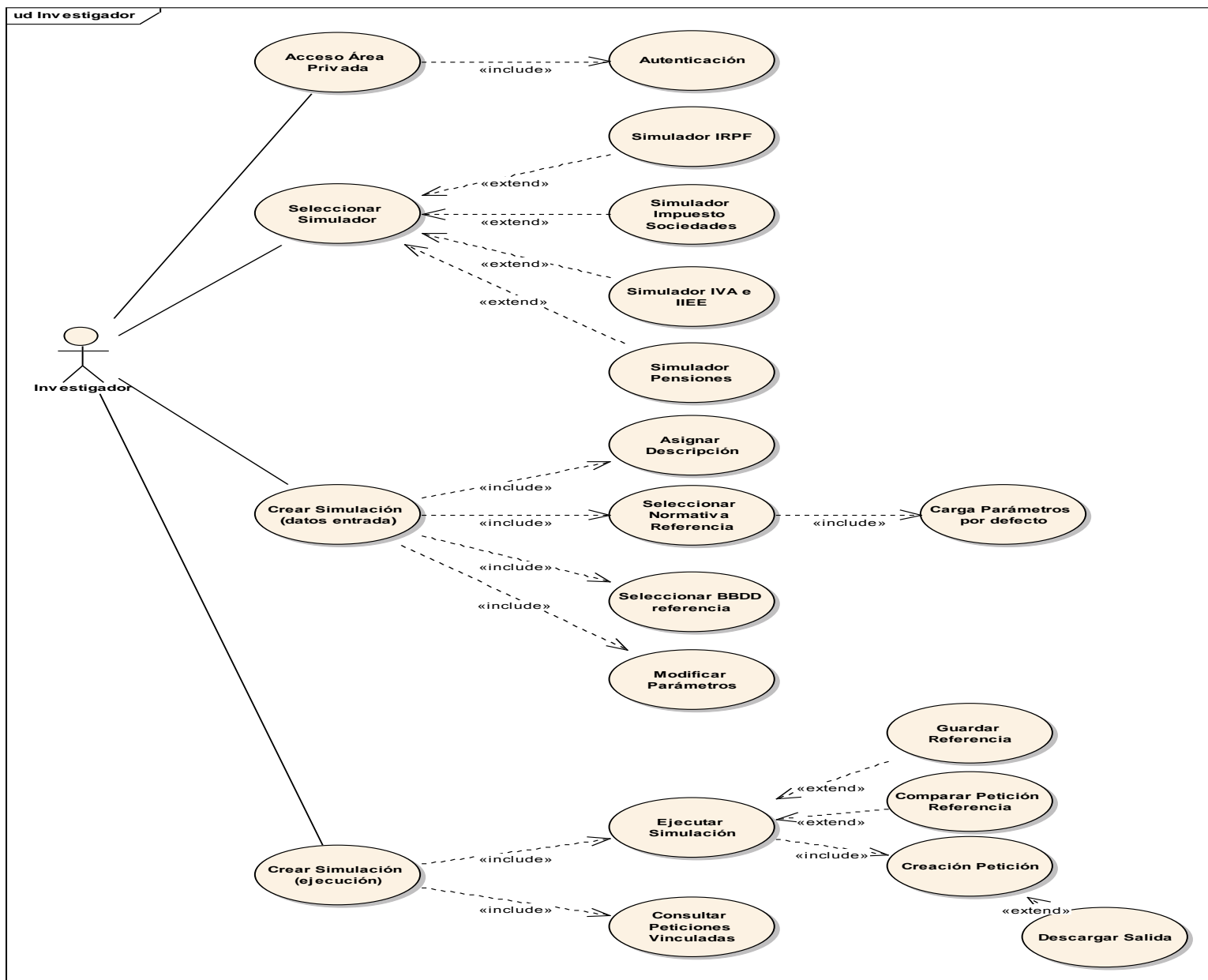
1. Introducción
2. Tipos de simuladores
 - I. Simuladores de IRPF
 - a) Objetivos.
 - b) Datos de entrada del simulador.
 - c) Ejecución del simulador.
 - d) Resultados
 - II. Simuladores del Impuestos de sociedades.
 - III. Simuladores IVA.
 - IV. Simuladores de pensiones.
3. Arquitectura de simuladores
 - I. Análisis funcional: casos de uso.
 - II. Descomposición en subsistemas.
 - III. Arquitectura lógica.
 - a) Interfaz web del simulador. Entorno de desarrollo.
 - b) Base de datos.
 - c) Servicios Web.
 - d) Módulo servidor. Servidor WF.
 - IV. Arquitectura física.

3.1 Análisis funcional: Casos de Uso (1)

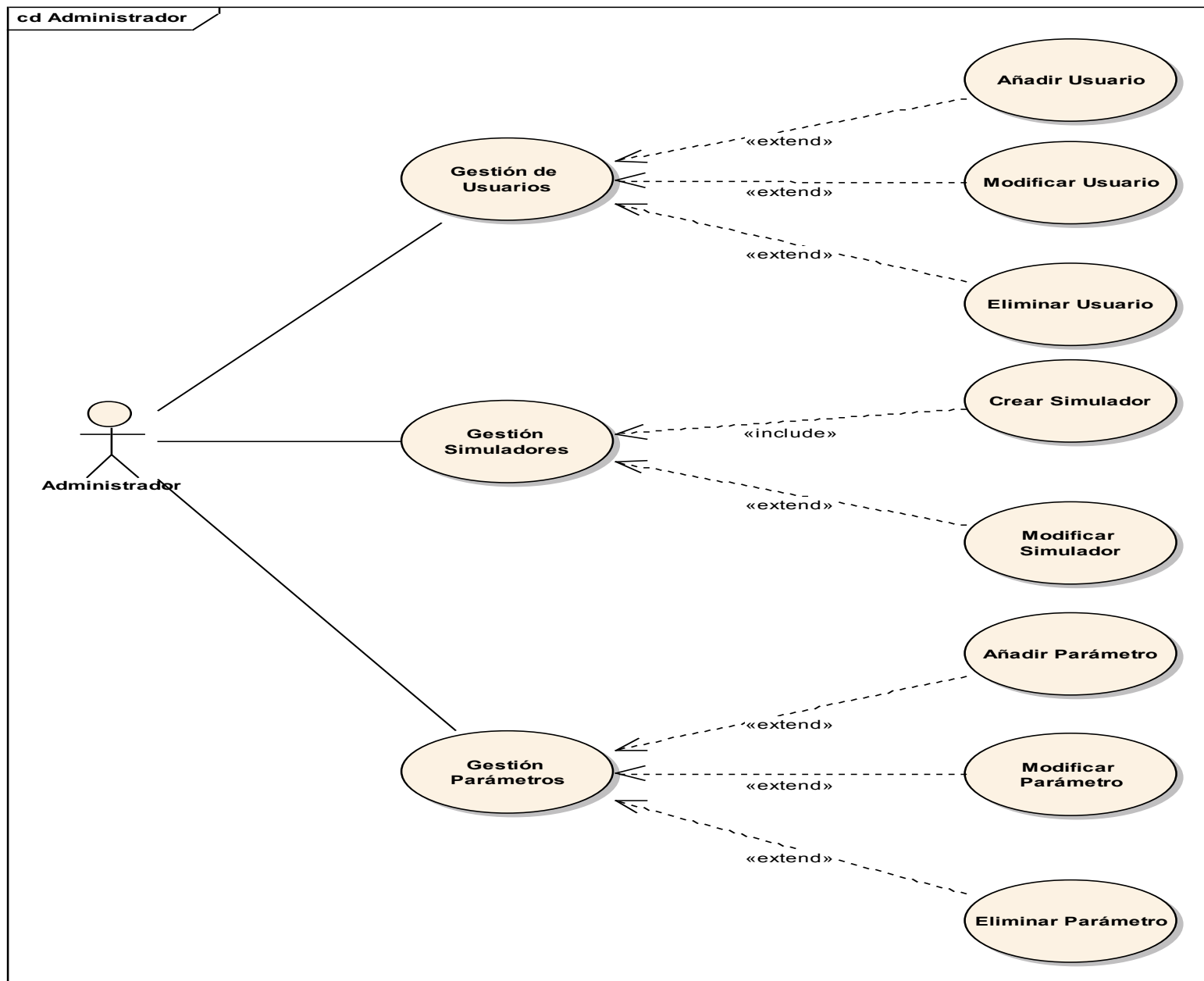
El **diagrama de casos** de uso sirve para **representar**, para cada uno de los **actores identificados**, qué **funciones** pueden realizar con el sistema.

- Las **relaciones** entre los casos de uso **pueden ser**:
 - **<<communicates>>** o <<comunica>>: **entre un actor y un caso de uso**. Se representa con una línea sin origen y final.
 - **<<includes>>** o <<usa>>: cuando **un caso de uso incluye la ejecución de otros casos** de uso. Se representa con una flecha con línea discontinua que va desde el caso origen al caso destino.
 - **<<extends>>** o <<extiende>>: es un **comportamiento opcional de un caso de uso**, es decir, que puede ejecutarse dentro de un caso de uso o no dependiendo de alguna condición. Se representa con una flecha con línea discontinua que va desde el caso de uso que es opcional hacia el caso de uso que es el origen o principal.

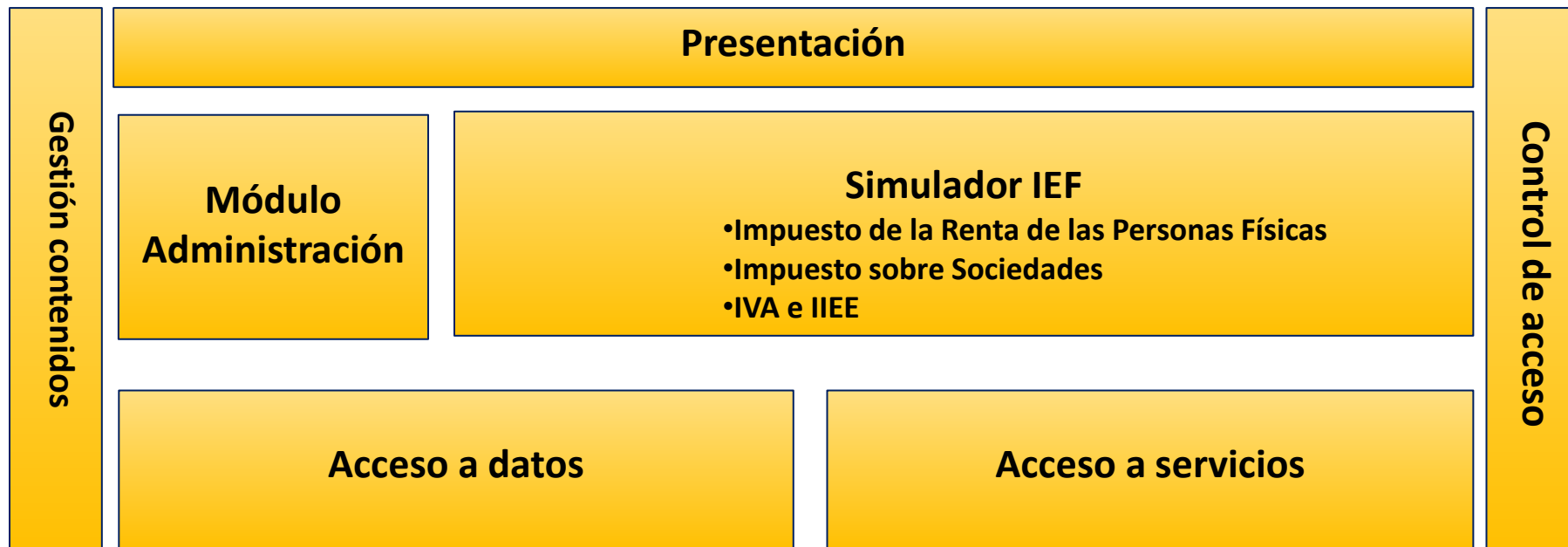
3.I Análisis funcional: Casos de Uso (2)



3.I Análisis funcional: Casos de Uso (3)



3.II Descomposición en subsistemas



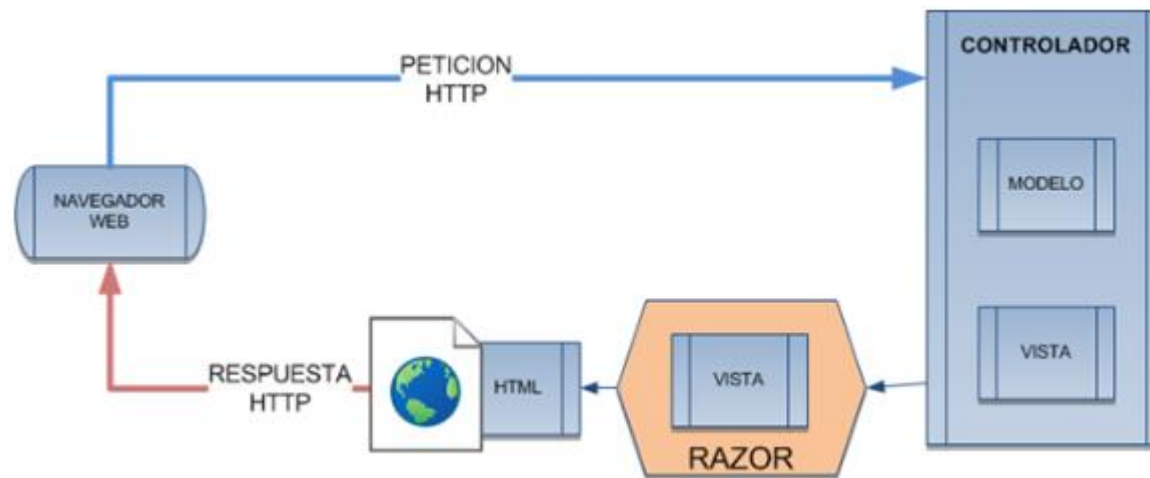
3.III Arquitectura lógica (1)

- La arquitectura del simulador web es una **arquitectura servidor multicapa**.
 - se caracteriza por **dividir** el **sistema** en **distintos niveles** o capas que **se comunican** a través de **interfaces**.
- Esta arquitectura **busca**:
 - **Aprovechar** las especiales **características** de los **diferentes lenguajes** o tecnologías de desarrollo.
 - **Independizar** unas **capas** de otras tratando de conseguir que **un cambio** en una capa **no afecte al resto** de capas del sistema.

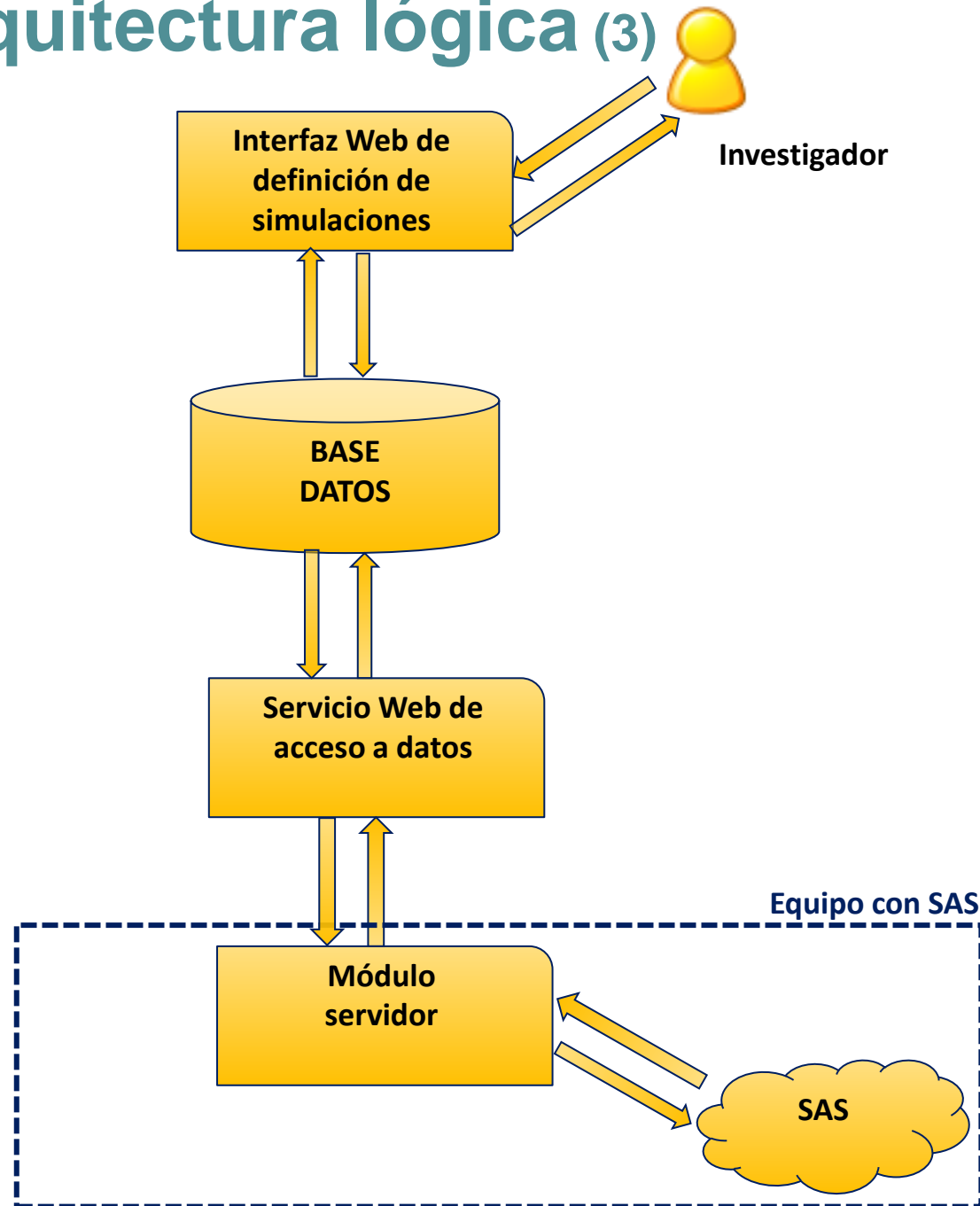
3.III Arquitectura lógica (2)

Está basado en un **Modelo Vista Controlador (MVC)** como patrón arquitectónico. Los elementos que componen el patrón son los siguientes:

- El controlador. Es el elemento encargado de recibir y procesar la peticiones HTTP. Su responsabilidad es retornar una vista concreta con el modelo correcto.
- El modelo. Corresponde con la representación de los datos/entidades de negocio que modelan la aplicación.
- La vista. La vista que ha retornado el controlador, es interpretada por el motor de renderización de ASP.NET MVC «Razor», que procesa la vista para generar el documento HTML que será devuelto finalmente al navegador

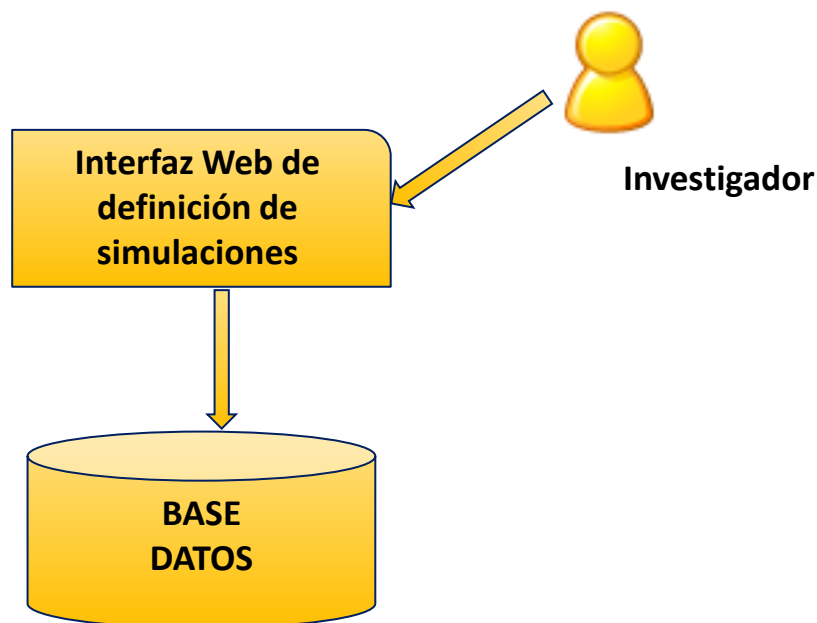


3.III Arquitectura lógica (3)



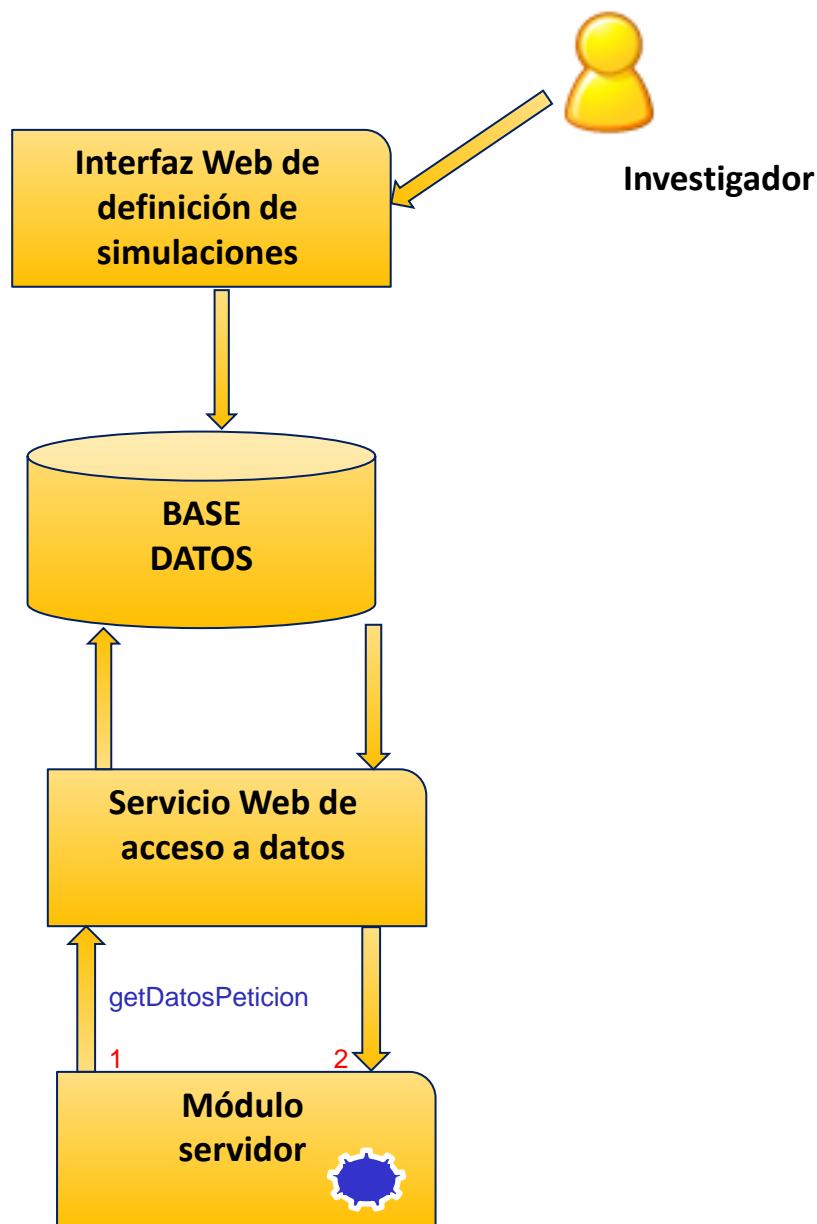


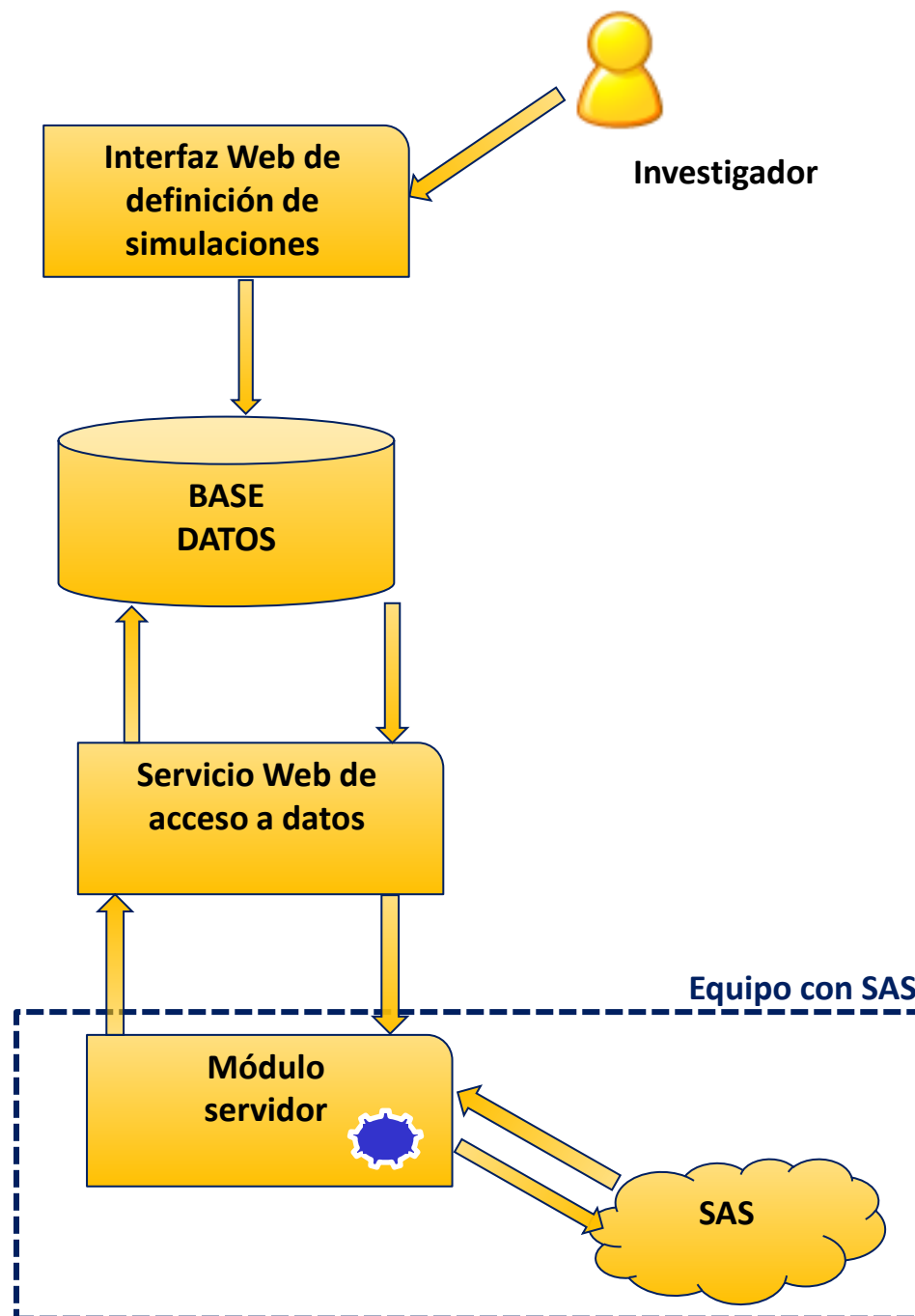
- Investigador accede al área privada (autenticación).
- Selecciona el simulador.
- Crea Simulación
 - Selección de BBDD y normativa
 - Modificación de parámetros.
- Ejecuta Simulación



- Se almacena la petición del usuario (parámetros de la simulación) en la BBDD.
- Los parámetros generales están pre-cargados en la BBDD.

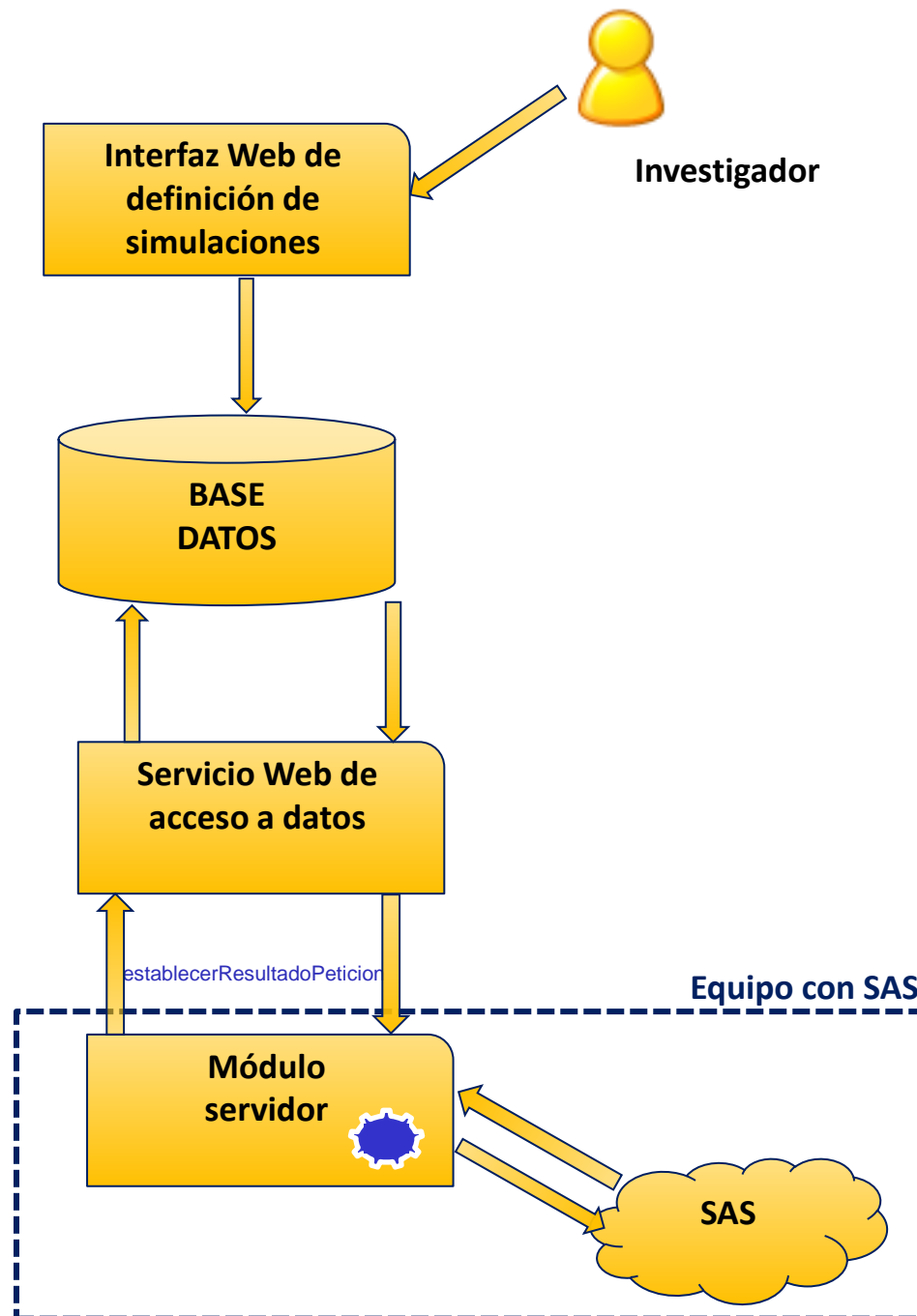
- El módulo servidor se encarga de consultar las peticiones que hay pendientes en la base de datos (servicio).
- Accede a la BBDD a través de servicio web para la lectura de los parámetros de la petición en cuestión.





- Si existe petición, el módulo servidor invoca el programa SAS una vez conocidos los parámetros.
- Realiza la simulación.

- Los resultados de la simulación se guardan en base de datos a través del servicio web.





1. Introducción
2. Tipos de simuladores
 - I. Simuladores de IRPF
 - a) Objetivos.
 - b) Datos de entrada del simulador.
 - c) Ejecución del simulador.
 - d) Resultados
 - II. Simuladores del Impuestos de sociedades.
 - III. Simuladores IVA.
 - IV. Simuladores de pensiones.
3. Arquitectura de simuladores
 - I. Análisis funcional: casos de uso.
 - II. Descomposición en subsistemas.
 - III. Arquitectura lógica.
 - a) Interfaz web del simulador. Entorno de desarrollo.
 - b) Base de datos.
 - c) Servicios Web.
 - d) Módulo servidor. Servidor WF.
 - IV. Arquitectura física.

3.III a Arquitectura lógica

Interfaz web. Entorno de desarrollo (1)

- Es la **encargada** de **mostrar la interfaz de usuario al investigador e interactuar** con él.
- La herramienta de desarrollo seleccionada es **Microsoft Visual Studio**:
 - ***Minimizar*** los ***riesgos*** de implantación y ***facilitar*** la ***integración*** en los sistemas del IEF (MINHAP).
 - ***El personal IEF cuenta con conocimiento en desarrollo con Visual Studio.***
- Además **Microsoft Visual Studio permite**:
 - **Desarrollo** de **Servicios Web**.
 - **Fácil interacción con SAS** mediante el **acceso** a **utilidades** del sistema operativo Windows o ejecución directa de procesos. Para la **integración con SAS**, se ha decidido que se utilice la **llamada directa a través de procesos**.

Interfaz web. Entorno de desarrollo (2)

■ Principal **reto: encontrar** un interfaz gráfico para **poder editar más de 1500 parámetros**. El rendimiento de un formulario con este número de variables sería poco usable y además tendría un rendimiento muy bajo. Solución:

■ Utilización de una **clasificación temática** de las **variables**, de forma que se muestren **grupos pequeños**, con unas 20 ó 30 variables, que son fácilmente visualizables en un sólo vistazo.

■ Validación de los parámetros en el formulario web.

■ El desarrollo de la aplicación web se ha realizado con **ASP.NET**:

■ ASP.NET es parte de .NET Framework, acceso a todas las funciones de este marco de trabajo.

■ Ejemplo: crear aplicaciones Web ASP.NET con cualquier lenguaje de programación .NET (Visual Basic, C#, ...) y utilidades de depuración .NET. Acceso a los datos con ADO.NET.

1. Introducción
2. Tipos de simuladores
 - I. Simuladores de IRPF
 - a) Objetivos.
 - b) Datos de entrada del simulador.
 - c) Ejecución del simulador.
 - d) Resultados
 - II. Simuladores del Impuestos de sociedades.
 - III. Simuladores IVA.
 - IV. Simuladores de pensiones.
3. Arquitectura de simuladores
 - I. Análisis funcional: casos de uso.
 - II. Descomposición en subsistemas.
 - III. Arquitectura lógica.
 - a) Interfaz web del simulador. Entorno de desarrollo.
 - b) Base de datos.
 - c) Servicios Web.
 - d) Módulo servidor. Servidor WF.
 - IV. Arquitectura física.

3.III b Arquitectura lógica

Base de datos

- Se encuentra **implementada** en el **Sistema Gestor de Base de Datos SQL Server**.
- Esta capa ofrece los siguientes **servicios**:
 - Acceso a la base de datos mediante la utilización de **LINQ**, que permite abstraer el modelo de datos a unas clases de negocio.
 - Facilita el desarrollo
 - Minimiza los errores
 - Asegura coherencia entre los distintos sistemas

1. Introducción
2. Tipos de simuladores
 - I. Simuladores de IRPF
 - a) Objetivos.
 - b) Datos de entrada del simulador.
 - c) Ejecución del simulador.
 - d) Resultados
 - II. Simuladores del Impuestos de sociedades.
 - III. Simuladores IVA.
 - IV. Simuladores de pensiones.
3. Arquitectura de simuladores
 - I. Análisis funcional: casos de uso.
 - II. Descomposición en subsistemas.
 - III. Arquitectura lógica.
 - a) Interfaz web del simulador. Entorno de desarrollo.
 - b) Base de datos.
 - c) Servicios Web.
 - d) Módulo servidor. Servidor WF.
 - IV. Arquitectura física.

3.III c Arquitectura lógica

Servicios WEB (1)

- **Actualmente** es la **tecnología** por **excelencia** para **integrar aplicaciones**.
- Un servicio web es **un sistema software diseñado** para soportar la **interacción máquina-a-máquina o sistema-a-sistema** a través de una **red** y de forma **interoperable**.
 - En el simulador IEF la ***interacción*** se da entre ***ServidorWF*** (Módulo Servidor) y la ***base de datos***.
 - Los servicios web han sido desarrollados en el framework de Visual Studio, concretamente con C#.
 - En la capa de Servicios Web deben recogerse todos los accesos a la base de datos que se invocan desde el Módulo Servidor de la aplicación.

3.III c Arquitectura lógica

Servicios WEB (2)

- Las **operaciones definidas** en los **Web Services** son:
- **Servicio siguientePetición**: **Devuelve** la siguiente **petición pendiente** de ser **ejecutada** en el **simulador**. Al devolver el nº de esta petición, automáticamente se marca como ya ejecutada.
- **Servicio getDatosPetición**: Este servicio **devuelve** los **valores concretos** de la **petición solicitada**.. Para cada valor devuelve el valor definido para esta petición y el valor por defecto para esta versión de la base de datos.
- **Servicio establecerResultadoPetición**: Este servicio permite guardar en base de datos el **resultado** de una **petición concreta**. Recibe los siguientes parámetros:
 - 1) Código de la petición a la que se desea establecer el resultado.
 - 2) Estado en el que queda la petición.
 - 3) Parámetro binario con el fichero de resultados.

Este fichero es el que se descargará el usuario cuando consulte el resultado de su petición.

1. Introducción
2. Tipos de simuladores
 - I. Simuladores de IRPF
 - a) Objetivos.
 - b) Datos de entrada del simulador.
 - c) Ejecución del simulador.
 - d) Resultados
 - II. Simuladores del Impuestos de sociedades.
 - III. Simuladores IVA.
 - IV. Simuladores de pensiones.
3. Arquitectura de simuladores
 - I. Análisis funcional: casos de uso.
 - II. Descomposición en subsistemas.
 - III. Arquitectura lógica.
 - a) Interfaz web del simulador. Entorno de desarrollo.
 - b) Base de datos.
 - c) Servicios Web.
 - d) Módulo servidor. Servidor WF.
 - IV. Arquitectura física.

3.III d Arquitectura lógica

Módulo Servidor. Servidor WF (1)

- El servidorWF ha sido **desarrollado** en el **framework de Visual Studio**, concretamente con **C#**. Contiene la lógica de negocio de la aplicación.
- Se **realiza** las siguientes **acciones**:
 - **ejecutarpeticion**: se accede a través de las operaciones definidas en el webservice (**getDatosPeticion(codPeticion)**) a la BD para la lectura de los parámetros de la petición en cuestión.
 - **generarSASParametros**: Se transforma los parámetros leídos anteriormente de base de datos en un fichero SAS.
 - **LanzarProceso**: se invoca el programa SAS (residente en la misma máquina).
 - **publicarResultado**: Accede a través de servicio web a la base de datos (**establecerResultadoPeticion**). Los resultados de la simulación se guardan en base de datos en el campo ficheroresultado de la tabla Peticion. Este campo es de tipo varbinary.

- El Servidor WF **invoca al Programa SAS** (Statistical Analysis System) **para realizar la simulación.**
- Se ha optado por **SAS** ya que:
 - **Gran potencia** en el manejo de **grandes cantidades de datos**
 - **Capacidad** para **elaborar estadísticos descriptivos** necesarios en las etapas iniciales de cualquier análisis o en cualquier análisis posterior.
 - **Capacidad para aplicar** técnicas multivariantes, previsión, modelización, programación lineal frente a lenguajes multipropósito.

1. Introducción
2. Tipos de simuladores
 - I. Simuladores de IRPF
 - a) Objetivos.
 - b) Datos de entrada del simulador.
 - c) Ejecución del simulador.
 - d) Resultados
 - II. Simuladores del Impuestos de sociedades.
 - III. Simuladores IVA.
 - IV. Simuladores de pensiones.
3. Arquitectura de simuladores
 - I. Análisis funcional: casos de uso.
 - II. Descomposición en subsistemas.
 - III. Arquitectura lógica.
 - a) Interfaz web del simulador. Entorno de desarrollo.
 - b) Base de datos.
 - c) Servicios Web.
 - d) Módulo servidor. Servidor WF.
 - IV. Arquitectura física.

■ Entorno tecnológico propuesto en dos niveles:

- Desde la **perspectiva** de **Arquitectura de Sistemas** se encuentran los siguientes elementos:
 - Internet Information Server como servidor frontal para atender las peticiones HTTP.
 - SAS Application server como Servidor de Aplicaciones.
 - Windows Server 2012: Sistema Operativo de referencia.
- Desde la **perspectiva** de **Arquitectura de Aplicaciones** se proponen la utilización de los siguientes frameworks de desarrollo:
 - Microsoft Visual Studio 2008.
 - SAS Base: Plataforma de modelación y desarrollo de procesos.

Arquitectura física. CPD (1)

Máquinas virtuales

- La plataforma del sistema está basada en **máquinas virtuales sobre Hyper-V**, Estas máquinas deben tener el **Sistema Operativo “Windows Server 2012 Standard para 64 bits”**.
- La máquina sobre la que se ejecuta el servidor de aplicaciones tiene las siguientes características:

Requisitos	Tipo	Necesidad
RAM	Hardware	50 GB
Sistema Operativo	Software	Windows Server 2012
Procesador ES-2640: Intel ® Xeon ® E5-2640 (15M Cache, 2.50 GHz, 7.20 GT / s Intel ® QPI) 6 núcleos y 12 hilos.		

Arquitectura física. CPD (2)

■ Se ha optado por servidores virtualizados ya que:

- Permiten **optimizar** el **reparto** de los **recursos** entre las plataformas y disponer de un alto grado de **tolerancia** ante **fallos**.
- Mediante la **virtualización**, también se **mejora** la capacidad de **gestión de los servidores y su seguridad**, una utilización de los recursos significativamente mayor mediante la agrupación de infraestructuras comunes y las facilidades para el **dimensionamiento dinámico** de las mismas.
- La consolidación y virtualización de los diferentes entornos y elementos de las plataformas conseguirá **reducir los costes** asociados al **mantenimiento del hardware**, crecimientos futuros, consumo eléctrico y espacio físico necesario.

Arquitectura física. CPD (3)

- La **virtualización** **permitirá** cubrir de forma más sencilla los posibles **nuevos requerimientos funcionales** o de carga que puedan producirse en el medio plazo, mediante nuevos recursos para los servidores virtualizados, como la **memoria RAM** o las **CPU's** adicionales, sin necesidad de inversiones.
- Mediante la **creación** de **nuevos servidores virtuales** sobre misma plataforma, se podrían **crear más entornos**, (formación, pruebas, etc) sin nuevos servidores.

Arquitectura física. CPD (4)

- La **máquina** sobre la que se **instale** el **servidor** de aplicaciones **da soporte** a tres **aplicaciones para**:
 - Lanzar el proceso SAS, que se trata de un servicio que consulta las peticiones que hay pendientes en la base de datos.
 - Alojar la aplicación web del Simulador IEF en el servidor de aplicaciones. Se va a tratar de una aplicación Intranet, accesible por un número limitado y no muy elevado de usuarios.
 - Proporcionar servicios web publicados en el servidor de aplicaciones.

Arquitectura física. CPD (5)

❖ Entorno Preproducción:

- - Se propone SAS Application Server 9.4. En cualquier caso, la versión más actual .
- - Espacio en disco sobreestimado, almacena software y ficheros estáticos.

Requisito	Tipo	Necesidad
Espacio en disco	Hardware	2 Tb
Servidor aplicaciones	Software	SAS App. Server 9.4

❖ Entorno producción

- - Instalar versión más reciente existente.
- - Habrá correspondencia entre las versiones del servidor de aplicaciones que se instalará en ambos entornos.

Requisito	Tipo	Necesidad
Espacio en disco	Hardware	6 Tb
Servidor aplicaciones	Software	SAS App. Server 9.4

Sistema Gestor de Base de Datos

- La base de datos está creada sobre un Sistema Gestor de Base de Datos **SQL Server Database 2008R2 Enterprise**.
- Debe **presentar** las **condiciones** necesarias de **escalabilidad** para poder **almacenar las peticiones de simulaciones recibidas**. Se debe tener en cuenta para su correcto dimensionamiento las peticiones anuales.

Almacenamiento

- El **SW** desplegado en este servidor es **intensivo** en **I/O**. Es crítico que el entorno de almacenamiento pueda proporcionar un nivel de I/O alto.
- **RAID 5** se caracteriza por su **alto rendimiento y alta disponibilidad**.
 - DISCOS: 2,5" - 10K rpm - 6Gb/s - 1,2 Tb (16 unidades) en RAID 5.
 - Anualmente: 300 Gb
 - Cada cinco años: 1,5 Tb.

**Asesoría especializada para el desarrollo de
herramientas de microsimulación fiscal**

Implementación y arquitectura de los simuladores en el IEF.

jaime.villanueva@ief.minhap.es