



## Conexión indolora de modelos socioeconómicos y biofísicos

Una herramienta escrita por Julien Malard

[www.github.com/julienmalard/tinamit](https://www.github.com/julienmalard/tinamit)

# Índice

<b>1</b>	<b>Introducción</b>	<b>4</b>
1.1	¿Qué?	4
1.2	¿Por qué?	4
<b>2</b>	<b>Instalación</b>	<b>6</b>
2.1	Si sabes lo que estás haciendo	6
2.2	Si no sabes lo que estás haciendo (para Windows)	6
2.2.1	Instalación de Python	6
2.2.2	Instalación de paquetes adicionales	6
2.2.3	Uso con PyCharm	7
<b>3</b>	<b>Uso</b>	<b>8</b>
3.1.1	Preparación del modelo DS	8
3.1.2	Preparación del modelo biofísico	8
3.2	IGU	8
3.2.1	Cambiar idiomas	8
3.2.2	Cargar modelos (I)	9
3.2.3	Conectar variables (II)	9
3.2.4	Simular (III)	9
3.2.5	Incertidumbre (IV)	9
3.2.6	Guardar y cargar modelos conectados	9
3.3	IPA	9
3.3.1	Preparar todo	9
3.3.2	Cargar modelos	10
3.3.3	Conectar variables	10
3.3.4	Simular	10
3.3.5	Implementación de políticas	11
3.3.6	Resumen	11
<b>4</b>	<b>Desarrollo</b>	<b>12</b>
4.1	Envolturas de modelos BF	12
4.2	Envolturas de modelos DS	12

4.3 Interfaz ..... 12

# 1 Introducción

## 1.1 ¿Qué?

En breve, Tinamit sirve para conectar modelos de dinámicas de los sistemas (DS) con modelos biofísicos. Permite el intercambio de valores de variables entre ambos modelos (y en ambas direcciones) a cada paso de simulación, y eso, con un mínimo de código (o, si de verdad lo quiere así, sin código cualquier).

## 1.2 ¿Por qué?

¿Por qué darse una tal tarea? Bueno, querer conectar modelos es algo bastante común. En nuestro caso, los modelos DS son excelentes para involucrar a actores claves en el desarrollo del modelo, porque son visuales e intuitivos (al menos, tanto como lo puede ser un modelo). Si uno está, por ejemplo, construyendo un modelo del sistema humano (socioeconómico) de agricultores con problemas de degradación de suelos, un modelo DS sería perfecto para involucrar a agricultores, investigadores, agentes del gobierno, y otros en la construcción del modelo socioeconómico.

Pero son bastante inútiles para representar procesos físicos más complejos, como el flujo de agua y de sal que resulta en la degradación del suelo. Esto es problemático, porque la relación entre humanos y ambiente es recíproca, y para obtener resultados interesantes tenemos que incluir los dos y sus interacciones. Por ejemplo, la calidad del suelo y el rendimiento agrícola determinará el ingreso de agricultores, lo cual en su turno determinará las prácticas de gestión de los suelos que adoptan.

Y sí, por supuesto que ya existen muchas maneras de conectar estos tipos de modelos. Pero la mayoría involucra cientos de líneas de código, que usted tendrá que entender (incluso, por ejemplo, el manejo del DLL de Vensim y de la línea de comanda), y, si jamás decide uno hacia cambiar el nombre de un variable o conectar un variable más (o menos), habrá que volver a meterse de nuevo a descifrar todo el código. Lo que no es un punto favorable cuando quieres convencer una profesional en, digamos, agricultura o política que el modelo que vas a construir con ella será muy fácil utilizar después de que te vayas.

Por eso se inventó Tinamit, el programa gratis y libre para hacer todo eso para ti. Viene con:

- Funciones predefinidas y flexibles para conectar modelos DS y biofísicos y simular el modelo conectado.
- Un interfaz gráfico de usuario (IGU) muy lindo, para los que le tienen miedo al código.
- Hablando del IGU, está disponible en varios idiomas, y siempre puedes agregarle una nueva y compartirla con quien quieras.

Con Tinamit, cambiar la conexión entre los modelos o lanzar una nueva simulación se hace con el pequeño cambio de una línea de código o, en el interfaz, con un clic del ratón.

Lo único difícil es que, para cada tipo de modelo biofísico, hay que escribir un código (una “envoltura”) que pueda llamar las funciones de simulación y de lectura y cambio de valores de variables en este modelo biofísico. Sí, es trabajo para alguien a quien le gusta la programación, pero lo bueno es que una vez hecha para un tipo de modelo biofísico ya no hay necesidad de volverlo a hacer, no importe con qué modelo lo vas a conectar. Tinamit ya viene con envolturas para algunos modelos biofísicos, y seguirá agregando más en tanto los escriben sus usuarios (justedes!). Bueno, sabemos que no es ideal, pero es

una limitación inevitable de conectar modelos (aunque no usaras Tinamit, lo tendrías que hacer de todo modo). Y, de verdad, para todo lo bueno de Tinamit, es un asunto de poca importancia.

En este manual, se supondrá que ya conoces, al menos, el siguiente:

- Modelos de dinámicas de los sistemas
- Modelos físicos (en general)
- Python

Si no es el caso, ahora sería un momento excelente para ir a estudiarlos un poco. (¿Sabías? Para Python, [www.codecademy.com](http://www.codecademy.com) tiene un curso fenomenal, gratis y en español.)

Diagrama del programa

## 2 Instalación

### 2.1 Si sabes lo que estás haciendo

¡Felicitaciones! Tinamit necesita Python 3.5+, NumPy, SciPy, y Matplotlib. Por el momento, se instala Tinamit sí mismo por cargar la versión más recién del <https://github.com/julienmalard/Tinamit>.

### 2.2 Si no sabes lo que estás haciendo (para Windows)

Esto es para Windows. Es un poco complicado, así que si tienes dificultades no dudes en pedirme ayuda ([julien.malard@mail.mcgill.ca](mailto:julien.malard@mail.mcgill.ca)) para que te ahorres los dolores de cabeza que yo ya pasé.

Si estás usando Linux, no tengo idea cómo funciona pero me han dicho que es bastante más fácil que para los pobres que todavía seguimos con Windows.

Si estás usando un Mac, no tengo idea cómo funciona y parece bastante más difícil que para un Windows.

#### 2.2.1 Instalación de Python

Primer que todo, hay que instalar Python. Puedes cargar la versión la más recién de aquí (<https://www.python.org/downloads>).

#### 2.2.2 Instalación de paquetes adicionales

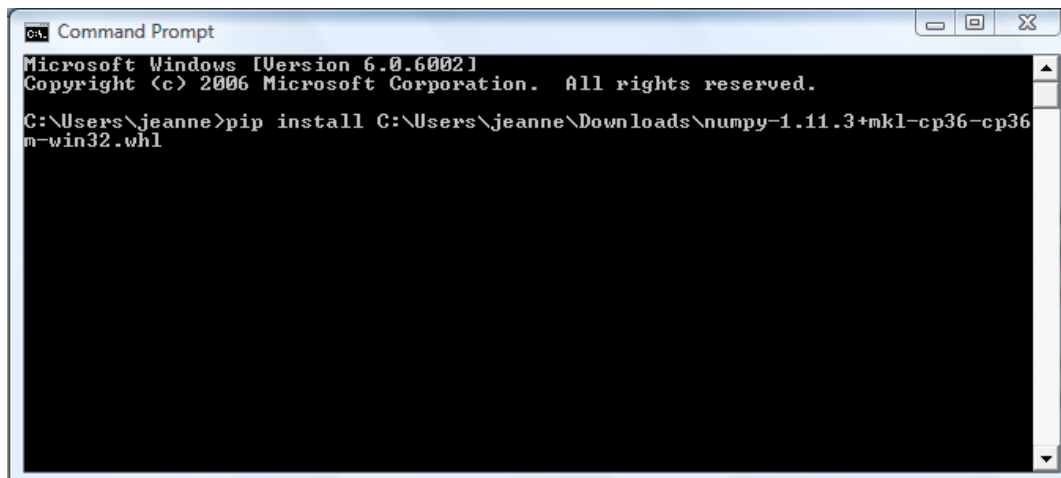
El problema con Python es que, mientras que es mucho más fácil para leer o escribir que otras lenguas (si no me crees, busca Fortran o C++), también es bastante más lento. Por eso, códigos Python que involucran muchos cálculos numéricos se escriben con extensiones en Fortran o en C para aumentar la velocidad un poco. No te preocupes, ¡que Tinamit no tiene nada en Fortran o C! Es puro Python. Pero desafortunadamente para sus funciones matemáticas necesita unos paquetes adicionales de Python (NumPy, SciPy y Matplotlib), y ellos, sí tienen extensiones raras.

Si sabes cómo instalar estas y compilar código C directamente en tu computadora, perfecto. Si, como la mayoría de la gente normal, no lo sabes, haz lo siguiente:

1. Ir al <http://www.lfd.uci.edu/~gohlke/pythonlibs>.
2. Descargar las versiones más recientes de [numpy+mkl](#), [scipy](#), y [matplotlib](#). Tienes que tener cuidado de escoger la buena versión para tu computadora. Todas estas tienen la forma general “nombreDelPaquete-versiónDelPaquete-versiónDePython-númeroDeBits.whl”. Asegúrate de descargar el archivo apropiado (si no sabes el número de bits de tu programa Python, es casi seguramente 32). Por ejemplo, para SciPy en Python 3.6 con un Python de 32 bits, escogerías “scipy-0.18.1-cp36-cp36m-win32.whl”.
3. ¡Bravo! Ahora, hay que instalar los paquetes que acabaste de cargar. Va a la *línea de comando* (se encuentra en la lista de aplicaciones de tu computadora) y escribe el siguiente:

```
pip install
C:\Users\jeanne\Downloads\numpy-1.11.3+mkl-cp36-cp36m-win32.whl
```

Después, pulse “Intro”. Por supuesto, tienes que cambiar la última parte para corresponder a dónde tú guardaste el paquete NumPy en tu computador cuando lo descargaste. Abajo está el ejemplo:



```
Command Prompt
Microsoft Windows [Version 6.0.6002]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.

C:\Users\jeanne>pip install C:\Users\jeanne\Downloads\numpy-1.11.3+mkl-cp36-cp36m-win32.whl
```

4. Repetir etapa 3 con con SciPy y después con Matplotlib.
5. ¡Casi terminado! Ahora, solamente tienes que ir a <https://www.microsoft.com/es-ES/download/details.aspx?id=53840> y descargar el “C++ 2015 redistributable” (toma la versión terminando en “...x86.exe” si tienes Python de 32 bits (si no lo sabes, toma este) y en “...x64.exe” si tienes Python de 64 bits. Después, instálalo. Por razones obscuras, SciPy no funciona en Windows sin este.

Bueno, si todo esto te parece un poco incómodo, estoy de acuerdo. Hay una nueva lengua de programación llamada [Julia](#) que es tan rápida como C y tan intuitiva como Python, y por lo tanto no tiene nada de extensiones ajenas en Fortran o en C. Pero me di cuenta demasiado tarde y ahora no voy a reescribir todo el programa de Tinamit en Julia (después de todo, tengo una tesis a escribir). Lo siento.

### 2.2.3 Uso con PyCharm

Para personas que piensan hacer más con Tinamit que usar el IGU, recomiendo muy fuertemente que usen la versión Comunitaria (gratis) de PyCharm (<https://www.jetbrains.com/pycharm/>). PyCharm es para Python lo que Word es para documentos de texto, y te salvará de muchos dolores de cabeza (por una cosa, te dice dónde has hecho un error).

Es bastante fácil usar PyCharm; después de instalarlo, simplemente hay que abrir la copia local de Tinamit en el editor y empezar a escribir tu código. Si quieres contribuir a Tinamit, puedes usar PyCharm para conectar tu versión con la página de Tinamit en GitHub (así siempre tendrás la versión más recién). Contáctame ([julien.malard@mail.mcgill.ca](mailto:julien.malard@mail.mcgill.ca)) si estás interesada.

## 3 Uso

Hay dos maneras de usar Tinamit. Si quieres hacerlo sin código cualquier, el IGU (Interfaz de Usuario Gráfico) es para ti. Si prefieres hacerlo con unas pocas líneas de código (lo cual puede acelerar bastante tu trabajo si tienes muchas simulaciones diferentes que quieres automatizar), entonces el IPA (Interfaz de Programación de Aplicaciones) es la mejor opción.

Al final, el IGU y el IPA hacen el mismo trabajo (el IGU tiene botones muy lindos para llamar las funciones del IPA de manera automática).

### 3.1.1 Preparación del modelo DS

De cualquier modo, tendrás que preparar el modelo de dinámicas de los sistemas antes de poder conectarlo. Para VENSIM (por el momento, Tinamit funciona con VENSIM solamente), primero hay que ir a cada variable en tu modelo VENSIM que quieres que pueda recibir valores desde el modelo biofísico y escoger “Gaming” como tipo de variable. Después, hay que publicar el modelo en formato .vpm.

### 3.1.2 Preparación del modelo biofísico

Si ya existe una envoltura específica para el modelo biofísico, no tienes que hacer nada más. Si no está disponible ya, tendrás que escribir una (o convencer a alguien de hacerlo para ti). La sección 4.1 abajo te explicará cómo hacer esto.

## 3.2 IGU

Para los que no quieren programar, el IGU ofrece una manera sencilla de acceder (casi) todas las funcionalidades del IPA Tinamit, y algunas adicionales.

### 3.2.1 Cambiar idiomas

Bueno, primero, de pronto no hables español. O posiblemente trabajas con gente que no lo habla, y quieres hacerles la cortesía de trabajar en su idioma. O tal vez quieres practicar un idioma que no has hablado por mucho tiempo.

Mientras que el código de Tinamit sí mismo es en español, el interfaz de Tinamit está disponible en muchos idiomas (y siempre puedes agregar un nuevo).

Para cambiar idiomas, hacer clic en el icono del globo terrestre.

En el centro, tienes las lenguas ya traducidas. Puedes escoger una con la cajita verde a la izquierda, o pulsar en lapicito para hacer cambios a la traducción.

A la izquierda, tienes lenguas en progreso. La barra muestra el estado del progreso de la traducción. Puedes escoger una como lengua de interfaz; las traducciones que faltan aparecerán como espacios vacíos. También puedes hacer clic en el lapicito para contribuir a la traducción.

Y, por fin, a la derecha tienes lenguas que todavía no hemos empezado a traducir. Puedes hacer clic en el lápiz para empezar la traducción. También puedes hacer clic en la cruz arriba para agregar un nuevo idioma que no se encuentra en la lista ya (y también especificar si se escribe de la izquierda hacia la derecha o al revés).



Todas las traducciones se guardan automáticamente en un documento llamado “Trads” en el directorio de Tinamit. Si contribuyes a unas traducciones, puedes compartir este documento ([julien.malard@mail.mcgill.ca](mailto:julien.malard@mail.mcgill.ca), o por GitHub) para que todas tengan acceso a tu idioma favorito.

### 3.2.2 Cargar modelos (I)

El flujo de trabajo en Tinamit tiene cuatro etapas (en números mayas) y el interfaz desbloquea el acceso a cada etapa en cuanto termines la etapa precedente.

La primera etapa sería, por supuesto de cargar los modelos biofísicos y DS.

### 3.2.3 Conectar variables (II)

Después de eso, vamos a conectar los dos modelos por sus variables comunes. La flecha muestra la dirección de la conexión, y puedes especificar un factor de conversión, si quieres. Hay que hacer clic en “guardar” cada vez que haces una conexión.

Si haces un error, puedes volver a editar una conexión ya hecha por hacer clic en el lapicito verde, o simplemente borrarla con la cruz roja. Nota que el interfaz no te dejará conectar un variable más que una vez al mismo tiempo (eso sería una falla lógica en la conexión de los dos modelos).

### 3.2.4 Simular (III)

Ya puedes simular los modelos conectados. Puedes especificar el paso y el tiempo final de la simulación. Finalmente, puedes especificar un factor de conversión entre el paso de cada modelo si los dos modelos no tienen las mismas unidades para sus pasos de tiempo (por ejemplo, si tu modelo DS funciona en meses y tu modelo biofísico en años, lo cual sería una situación muy común).

### 3.2.5 Incertidumbre (IV)

Un día, en el futuro, Tinamit tendrá unas funciones de autocalibración y de análisis de incertidumbre. Entre tanto, la página para esta cuarta etapa queda un blanco muy bonito.

### 3.2.6 Guardar y cargar modelos conectados

Ah, sí, ¿qué pasa si no terminas todo antes del almuerzo? No te preocupes, que no vas a perder todo. Allí, arriba por a la izquierda del logo muy bonito de Tinamit, hay cuatro botones muy útiles. Uno guarda tu trabajo, uno lo guarda bajo un nuevo nombre, otro abre un trabajo ya guardado y el último borra todo y te deja empezar de cero. No te voy a decir cuál es cuál.

## 3.3 IPA

El IPA (interfaz de programación de aplicaciones) permite conectar modelos de manera rápida, flexible, y reproducible. Tiene las mismas funciones que el IGU, pero por ser una librería Python ya puedes automatizar el proceso. Si prefieres escribir líneas de código a hacer clic en botones, el IPA es para ti.

### 3.3.1 Preparar todo

Antes que todo, hay que importar los objetos de Tinamit que vamos a necesitar:

```
from Conectado import Conectado
```

Esta línea importa la clase *Conectado* del módulo *Conectado* de Tinamit. Increíblemente, es la única cosa que tenemos que importar.

### 3.3.2 Cargar modelos

Primero, vamos a empezar por crear una instancia de la clase *Conectado*. Si no sabes lo que es una instancia de una clase, o puedes simplemente copiar el texto abajo, o (mejor) puedes echarle otro vistazo a tu último curso en Python.

```
modelo = Conectado()
```

¿Pero cómo especificamos cuáles modelos biofísico y DS queremos? Esto se hace en la línea siguiente:

```
modelo.estab_mds("C:\\SahysMod\\julien\\GBSDM_V4.vpm")
```

estab\_mds, como probablemente adivinaste, establece el modelo DS. Le tienes que dar como argumento la ubicación del archivo .vpm de tu modelo DS publicado por VENSIM. En el futuro, si Tinamit puede aceptar modelos de otros programas que VENSIM, podrás poner otros tipos de archivos aquí. (Notar que habrá que cambiar la ubicación aquí según tu computadora.)

Y, para el modelo biofísico, especificamos la ubicación de la envoltura específica para el modelo biofísico que queremos usar. En este caso, vamos a usar SAHYSMOD, un modelo de flujos de agua subterránea y de salinidad. Esto no cambia mucho; cada vez que quieres conectar un modelo DS con un modelo en SAHYSMOD darás la misma envoltura, no importe cuáles variables estás conectando.

```
modelo.estab_bf(os.path.join(os.path.split(__file__)[0],  
                             'SAHYSMOD_wrapper.py'))
```

(No te preocupes por lo del `os.path.split(__file__)[0]`, es simplemente una manera en Python de obtener la dirección en tu computadora del directorio actual. Esto le permite al programa encontrar la envoltura para SAHYSMOD que viene con Tinamit, no importe dónde guardaste el programa Tinamit en tu computadora.)

### 3.3.3 Conectar variables

Ahora, vamos a conectar los dos modelos por crear enlaces entre los variables de cada uno. Cada conexión entre dos variables necesita 3 cosas: los nombres de los dos variables para conectar y la dirección de la conexión (es decir, de cuál modelo sacas el valor del variable para ponerlo en el otro modelo). Una simulación verdaderamente dinámica incluirá conexiones en ambas direcciones (del modelo DS al biofísico y viceversa).

```
modelo.conectar(var_mds='Soil salinity AS1', mds_fuente=False,  
                var_bf="Cr4 - Fully rotated land irrigated root zone salinity")
```

var\_mds es el nombre del variable en el modelo DS, y var\_bf es el nombre del variable en el modelo biofísico (tal como especificado en la envoltura). mds\_fuente indica si se lee el valor del variable en el modelo DS para transferirla al modelo biofísico, o si es al revés. En este ejemplo, tomamos el valor de la salinidad del suelo del modelo SAHYSMOD y lo pasamos al modelo DS (VENSIM).

Opcionalmente, puedes especificar el parámetro conv, un factor de conversión (si los dos variables tienen unidades distintas).

Puedes conectar tantas variables como quieras. En nuestro ejemplo, conectamos un total de 8 variables.

### 3.3.4 Simular

Ya, por fin, podemos simular el modelo:

```
modelo.simular(paso=1, tiempo_final=240, nombre_simul=name)
```

Paso indica el intervalo de tiempo al cual se intercambian valores de variables entre los dos modelos. tiempo\_final indica la duración de la simulación, y nombre\_simul es el nombre de la simulación que se dará al archivo con los egresos (resultados) de la simulación.

### 3.3.5 Implementación de políticas

Los que conocen los modelos de dinámicas de los sistemas sabrán que muchas veces se incorporan opciones de acciones o de políticas en los modelos con un variable “sí o no.” Por ejemplo, en nuestro modelo de salinidad de los suelos tenemos un variable llamado “Política de recuperación de aguas.” Si este variable es igual a 1, activará la parte del modelo de dinámicas de los sistemas para una política de recuperación de aguas. Si es igual a 0, no habrá intervención en la simulación. Estas cosas son muy útiles para comprobar la eficacia (o no) de varias ideas de intervenciones en el sistema.

El asunto es que nosotros nos aburrimos muy rápido de tener que cambiar los valores de estos variables en el modelo VENSIM y tener que volver a publicarlo cada vez que querríamos analizar una combinación diferente de políticas. Con 5 posibilidades de políticas distintas en nuestro modelo ejemplo, ¡acabamos con muchas combinaciones y permutaciones!

Así que no se preocupen, ya incluimos una función en el IPA que les permite activar o desactivar una política en particular si tener que abrir VENSIM y republicar el modelo para cada cambio. Actualmente, puedes emplear esta función para cambiar el valor de cualquier variable en el modelo antes de empezar la simulación, pero es más útil que todo para activar y desactivar políticas. (¡Cuidado! Esta función solamente cambia el valor inicial del variable.)

```
modelo.mds.cambiar_var(var="política maravillosa", val=1)
```

modelo.mds accede al objeto de modelo DS asociado con el modelo conectado, y la función .cambiar\_var() hacer exactamente lo que piensas que hace.

### 3.3.6 Resumen

Y bueno, allí está. Ya puedes conectar, desconectar, simular y manipular modelos. Mira el documento “Ejemplo SAHYSMOD” en el directorio de ejemplos de Tinamit para un ejemplo del uso del IPA en la automatización de corridas para simular, de una vez, 5 corridas de un modelo socioeconómico DS con un modelo biofísico de calidad y salinidad de los suelos (SAHYSMOD).

Para las que conocen las funciones *threading* de Python, y que piensan que sería una manera brillante de correr las 5 simulaciones en paralelo para ahorrar tiempo, no lo hagan. Pensamos lo mismo y cuando lo intentamos sucede que el DLL de VENSIM no puede correr más que un modelo al mismo tiempo y se pone en un gran lío. Si no tienes ni idea de lo que estoy diciendo, perfecto.

## 4 Desarrollo

Con las secciones anteriores, ya puedes instalar y usar todas las funcionalidades de Tinamit. Esta sección es para las personas que quieren contribuir a añadir nuevas funcionalidades y posibilidades para los otros usuarios de Tinamit.

### 4.1 Envolturas de modelos BF

### 4.2 Envolturas de modelos DS

### 4.3 Interfaz