

# TOBA

**Ambiente de Desarrollo Web**

---

*Marcia Tejada*

*Martin Arin*

## Índice

- [¿Qué es SIU-TOBA?](#)
- [Flujo de trabajo](#)
- [Esquema de Directorios](#)
- [Esquema de Ejecución](#)
- [Administración Básica](#)
- [Pasos de la instalación](#)
- [Creando ABMs.](#)
- [El código](#)
- [Ejemplo de código:](#)
- [Consola de comandos](#)
  - [Instalación](#)
  - [Instancia](#)
  - [Proyecto](#)
- [Referencias:](#)

## ¿Qué es SIU-TOBA?

SIU-Toba es un Ambiente de Desarrollo Web que utiliza a PHP como lenguaje de programación. Lo llamamos Ambiente porque es una suite de distintas utilidades:

- Un conjunto de librerías que son consumidas en ejecución.
- Un editor web destinado a la definición/configuración del proyecto, creación de operaciones y definición de sus comportamiento.
- Un conjunto de comandos de consola destinados a administrar los proyectos creados con la herramienta.

Las definiciones realizadas en el editor web se las denomina metadatos, y junto a las definiciones en código conforman el comportamiento del proyecto creado. Durante el desarrollo estos metadatos son almacenados en una base de datos relacional denominada instancia.

Con Toba se han desarrollado aplicaciones de gran importancia para todas las universidades argentinas. Entre ellas, el SIU-Guaraní.

## Flujo de trabajo

El flujo de desarrollo con la herramienta podría definirse así:

1. Se utiliza el editor web de toba para definir una operación, sus pantallas, sus componentes gráficos, tablas que se consumen, etc. Todo esto se almacena en metadatos en una base de datos.
2. Se utiliza un editor PHP para crear el código necesario para cubrir lógica particular de la operación.
3. Durante este proceso se va probando la operación desde el mismo editor web haciendo ajustes contextuales.
4. Una vez terminada se utilizan los comandos administrativos para exportar el proyecto desde el puesto de desarrollo e importarlo en el sistema en producción.
5. En el sistema en producción sólo necesita las librerías o runtime para ejecutar el proyecto (código + metadatos).

## Esquema de Directorios

## SIU-TOBA - Ambiente de Desarrollo Web

La siguiente es una lista de los directorios más importantes de Toba y sus funcionalidades a alto nivel:

- bin: Contiene la puerta de entrada a los comandos de consola. Para poder ejecutarlos desde cualquier terminal/consola, a este directorio es necesario incluirlo en el PATH del sistema operativo.
- doc: Contiene documentación interna del proyecto. Para el desarrollador, la mejor documentación se encuentra en el wiki y en este tutorial.
- instalación: Contiene toda la configuración local (base que se utiliza, proyectos que se editan, alias de apache, etc.) y los metadatos locales (logs, usuarios, etc.). Generalmente es un directorio que no se versiona ya que solo contiene información local de esta instalación.
- php
  - 3ros: Librerías externas utilizadas en el proyecto.
  - consola: Código fuente de los comandos administrativos de consola.
  - contrib: Código contribuido por los proyectos, que aún no pertenecen al núcleo pero que esta bueno compartir.
  - lib: Clases sueltas propias comunes a todo el ambiente
  - modelo: Contiene una serie de clases que utilizan el editor y los comandos para editar metadatos y código. Forman una base útil para armar otras herramientas consumiendo una API de alto nivel. Por ejemplo si el proyecto determina que es necesario desarrollar un instalador con prestaciones extras, es un buen comienzo consumir estas clases.
  - núcleo: Runtime o conjunto de clases que se utilizan en la ejecución de un proyecto. La documentación de estas clases se encuentra publicada.
- proyectos: Este directorio contiene los proyectos propios del ambiente y es el lugar sugerido para nuevos proyectos. Aunque pueden situarlos en cualquier directorio, si están aquí es más fácil configurarlos.
  - mi\_proyecto:
    - metadatos: Contiene la última exportación de metadatos del proyecto.
    - php: Directorio que será parte del include\_path de PHP, se asume que el proyecto almacenará aquí sus extensiones y demás código.
    - temp: Directorio temporal no-navegable propio del proyecto
    - www: Directorio navegable que contiene los puntos de acceso a la aplicación.
    - css: Plantillas de estilos CSS del proyecto.
    - img: Imágenes propias del proyecto.
    - temp: Directorio temporal navegable del proyecto.
- temp: Directorio temporal no-navegable común.
- var: Recursos internos a Toba.
- www: Directorio navegable que contiene recursos web que consumen el runtime y los proyectos.
  - css: Plantillas de estilos CSS disponibles.
  - img: Imágenes comunes que pueden utilizar los proyectos.
  - js: Clases javascript propias de toba y externas.
  - temp: Directorio temporal navegable común.

Toba permite una completa customización de los sistemas en cuanto a colores y diseños por medio de plantillas css.

## Esquema de Ejecución

A diferencia de una librería clásica, no existe el concepto de procedimiento principal o main en el cual el programador incluye las librerías y las consume. En Toba la situación es distinta:

- El proyecto brinda un punto de acceso en donde se incluye al núcleo de toba (generalmente es el archivo `www/aplicacion.php`).
- A partir de allí el núcleo analiza los metadatos del proyecto y de la operación puntual que se

ejecuta, activando los componentes acordes.

- Si alguna clase del runtime o algún componente se encuentra extendido por el proyecto, recién allí el programador puede incluir código propio, siempre enmarcado en un 'plan maestro' ya delineado.

Lo más interesante para mostrar en este tutorial es cómo el proyecto puede variar el comportamiento en ejecución.

## Administración Básica

Cuando utilizamos el editor web de Toba, estamos definiendo un proyecto en base a metadatos, almacenados en una base de datos definida durante la instalación.

Lo positivo de esto es que, al estar centralizada, es posible que un grupo de desarrollo localizado en la misma red pueda desarrollar sobre esta base en forma simultánea. Además se puede utilizar SQL tanto para manipular como para obtener los metadatos.

Lo negativo es que mientras estos metadatos no sean exportados al sistema de archivos no podrán ser compartidos con otros grupos de trabajo o dentro de un mismo grupo geográficamente distante. Esta necesidad de importar - exportar metadatos se cubre usando los comandos de consola. Como introducción a estos comandos necesitamos presentar dos:

- toba instancia exportar: Exporta desde la base hacia el sistema de archivos
- toba instancia regenerar: Importa desde el sistema de archivos hacia la base

## Pasos de la instalación

### Guía GNU/Linux

1. En caso de que los repositorios no tengan una versión actualizada de PHP (5.1.2 o superior) cambiar las fuentes de los paquetes.
2. Ejecutar como superusuario:  

```
apt-get install apache2 libapache2-mod-php5 php5-cli php5-pgsql php5-gd php5-mbstring
```
3. Ejecutar `php -m`. En caso de no encontrar los paquetes `pdo` y `pdo_pgsql` en el listado, se necesita instalarlos manualmente:
  - a. Descargar y compilar las extensiones utilizando el comando `pecl`:
  - b. 

```
apt-get install php5-dev php-pear postgresql-server-dev-8.1 build-essential  
pecl install pdo  
pecl install pdo_pgsql
```
  - c. Agregar a los archivos `/etc/php5/cli/php.ini` y `/etc/php5/apache2/php.ini`:
  - d. 

```
extension=pdo.so  
extension=pdo_pgsql.so
```
4. Para instalar servidor y cliente de Postgresql ejecutar (en caso de no contar con un servidor aparte)  

```
apt-get install postgresql
```
5. Para instalar el cliente Subversion ejecutar  

```
apt-get install subversion
```
6. Para instalar graphviz ejecutar (opcional)  

```
apt-get install graphviz
```

## SIU-TOBA - Ambiente de Desarrollo Web

7. Editar el archivo `/etc/php5/apache2/php.ini` y `/etc/php5/cli/php.ini`:

```
#Mínimos
magic_quotes_gpc = Off
magic_quotes_runtime = Off

#Recomendados
error_reporting = E_ALL      #Solo para desarrollo
display_errors = On         #Solo para desarrollo
memory_limit = 128M
post_max_size = 8 M
upload_max_filesize = 8 M
```
8. Descomprimir o bajar el código fuente de SIU-Toba por ejemplo en el path `~/proyectos/toba`
9. En caso que se haya descargado de un `.zip`, asignarle permisos de ejecución a los scripts de 'bin':

```
chmod +x ~/proyectos/toba/bin/*
```
10. Ejecutar y seguir las instrucciones:

```
~/proyectos/toba/bin/instalar
```
11. Tener en cuenta el esquema de permisos de UNIX. Apache necesita acceso a las carpeta `www` de Toba y del Proyecto. Si se quieren guardar los logs (opción por defecto) también necesita acceso de escritura a la carpeta instalación. Existe una comando que acomoda los permisos luego de la instalación y ejecución: (requiere ejecutarse como usuario privilegiado):

```
sudo ./bin/toba instalacion cambiar_permisos -u <mi_usuario> -g <Usuario Apache generalmente es www-data>
```
12. Para editar el proyecto navegar hacia `http://localhost/toba_editor/2.3`
13. Para ejecutar el proyecto navegar hacia `http://localhost/miproyecto/1.0` (cambiar la url según el identificador que ha dado al proyecto).

## Creando ABMs.

Toba tiene una interfaz amigable para crear vistas de cuadros, filtros y alta de operaciones.



Cuenta con un asistente para crear directamente ABMs .



Los ABMs estarán relacionados con alguna tabla de la base de datos asociada al proyecto.

ABM simple

Información básica Cuadro Formulario

Carpeta archivos (\*) abm\_socios

Anexo nombre clases (\*) \_abm\_socios

Fuente de Datos (\*) Fuente videoclub

Tabla (\*) socio

Puede filtrar los datos

Definición de campos

Campo	Etiqueta	Clave	Obligatorio	Tipo	Cuadro	Form.
nombre	Nombre	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Cadena	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
apellido	Apellido	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Cadena	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
id	Id	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Entero	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Recargar

Anterior Generar

Las fuentes de datos pueden ser varias, y podemos seleccionar cualquiera de ellas. También es posible filtrar datos en los cuadros, y Toba nos da la posibilidad de hacerlo automáticamente.

En el cuadro de abajo vemos cómo las columnas de la tabla seleccionada pasan a formar parte de nuestro formulario, pudiendo elegir si queremos que aparezcan en el cuadro, si serán claves (a la hora de seleccionar alguna fila en el cuadro) y si son obligatorias para el formulario.

ABM simple

Información básica Cuadro Formulario Filtro

Ocultar cuadro hasta que se presione BUSCAR

Requiere ingresar al menos un criterio

Opciones de búsqueda

Columna	Operador de comparación
nombre	ILIKE

Generar

Las opciones del filtro son para ver cómo vamos a filtrar. Ya que elegimos solamente por nombre,

## SIU-TOBA - Ambiente de Desarrollo Web

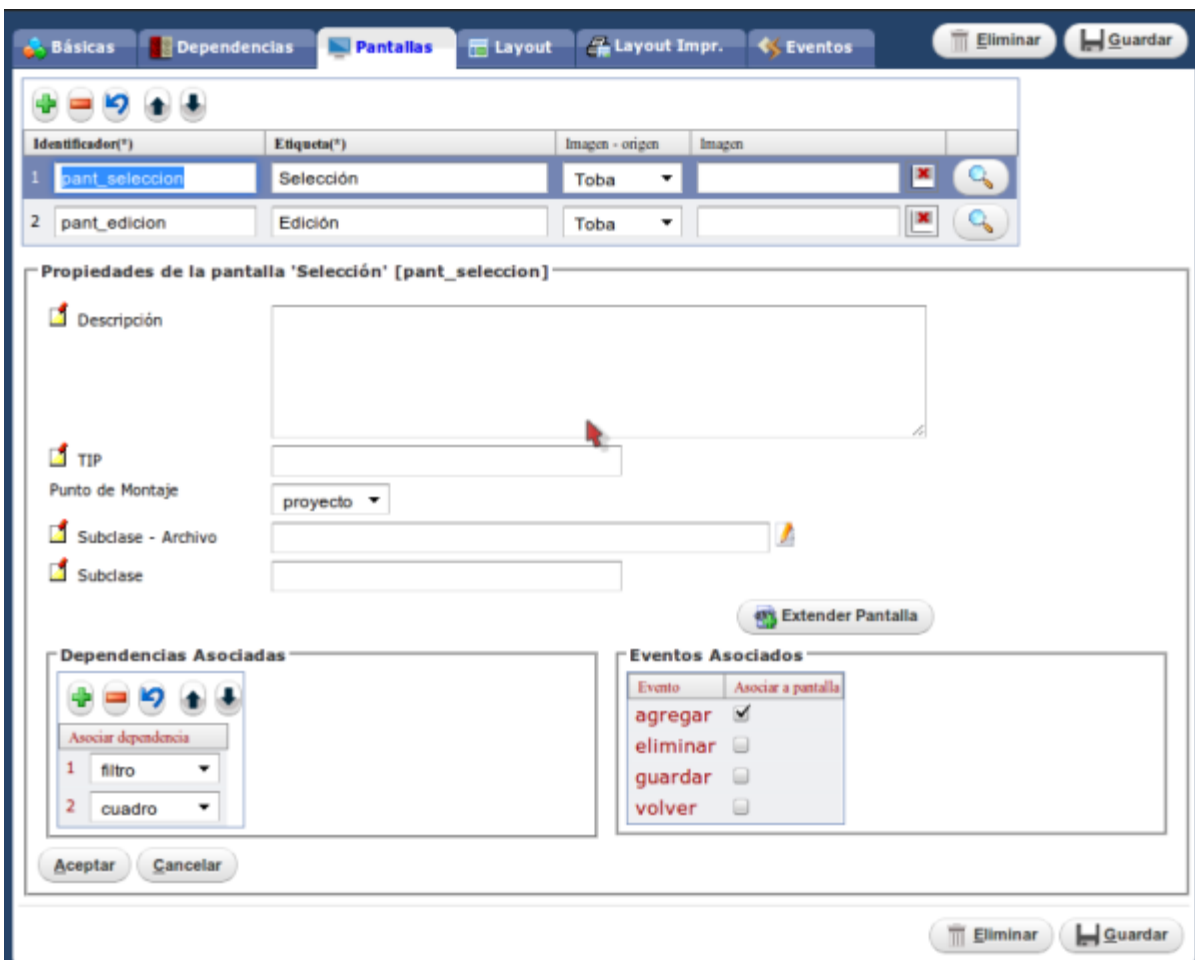
nos pregunta el operador para buscar, podemos cambiar por LIKE también.

En la pestaña de cuadro se genera la consulta que queremos que se realice a la base para llenar las filas.



Una vez que terminamos con el asistente, así es como queda la estructura del ABM.

En la parte de datos, nos crea una especie de controlador que nos brinda operaciones para poder trabajar de una manera más amena con la tabla que elegimos. La tabla "socio" queda como una dependencia de nuestra pantalla a la que podemos acceder mediante el código.





## SIU-TOBA - Ambiente de Desarrollo Web

Vemos que el ABM quedó dividido en dos pantallas.

En las “dependencias asociadas” se elige cuáles van a ser las que aparezcan en cada pantalla.

Los “eventos asociados” son los botones que van a aparecer en la pantalla. Al seleccionarlos es posible generar el código que va a manejar el evento.

The screenshot shows the 'Eventos' configuration screen in the SIU-TOBA web development environment. The interface includes a navigation bar with tabs for 'Básicas', 'Dependencias', 'Pantallas', 'Layout', 'Layout Impr.', and 'Eventos'. The main content area is titled 'Definir los eventos disponibles en el ci. Recuerde asociar a cada pantalla sus eventos.' and contains a table of event configurations. Below the table is a 'Generador de eventos' section with a 'Modelo de eventos' dropdown set to 'Guardar - Cancelar' and a 'Generar' button. At the bottom right, there are 'Eliminar' and 'Guardar' buttons.

Identificador(*)	Etiqueta	Imagen - origen	Imagen	Datos	en Bot.	Impl.	Predet.	
agregar	Agregar	Toba	nucleo/agregar.gif	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
volver	Volver	Toba	deshacer.png	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
eliminar	Eliminar	Toba	borrar.png	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
guardar	Guardar	Toba	guardar.gif	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

En la pestaña de eventos podemos crear todos los que creamos necesarios. Estos que vemos en la captura son los eventos en general del controlador de nuestra interface de ABMs.

Se pueden crear eventos por separado para cada dependencia (cuadros, filtros, formularios, etc.). Si le decimos a un evento que maneje datos (tilde en la columna “Datos”), cuando se realice esa llamada, al código se le pasarán los datos ingresados en un formulario (o seleccionados en un cuadro, por ejemplo).

Los cuadros pueden ser exportados a PDF y hojas de cálculo de manera automática.

Así es como queda el filtro y el cuadro:

The screenshot shows the 'ABM Socios' interface. It features a 'Filtro de Socios' section with input fields for 'Nombre' and 'Apellido', and a 'Filtrar' button. Below this is a 'Cuadro de Socios' section with a table of data. The table has columns for 'Id', 'Nombre', and 'Apellido'. A single row is visible with 'Id' 1, 'Nombre' Marcia, and 'Apellido' Tejada. There are also 'Eliminar' and 'Agregar' buttons at the bottom of the interface.

Id	Nombre	Apellido
1	Marcia	Tejada

ABM Socios

Agregar/Editar un socio

Id (\*)

Nombre (\*)

Apellido (\*)

## El código

Toba está escrito en PHP y es el lenguaje que vamos a usar para controlar las interfaces. Las pantallas pueden funcionar con AJAX y JavaScript para las validaciones (esto hace más ameno el uso del sistema ya que hace validaciones en tiempo real del lado del cliente sin tener que ir al servidor).

Tiene un generador de código y un editor de código internos.



Nos permite generar los eventos para cada dependencia (cuadro, filtro, etc) de nuestra pantalla.

## Ejemplo de código:

```
//---- Cuadro -----
/*
Configuración del cuadro.
*/
function conf__cuadro(toba_ei_cuadro $cuadro)
{
    if (isset($this->s__datos_filtro)) {
        $cuadro->set_datos($this->dep('datos')->tabla('Socio')->get_listado($this->s__datos_filtro));
    } else {
        $cuadro->set_datos($this->dep('datos')->tabla('Socio')->get_listado());
    }
}

/*
Comportamiento cuando se presiona el botón Eliminar.
El parámetro $datos son los datos de la fila seleccionada en el cuadro.
*/
function evt__cuadro__eliminar($datos)
{
    $this->dep('datos')->resetear();
    $this->dep('datos')->cargar($datos);
    $this->dep('datos')->eliminar_todo();
    $this->dep('datos')->resetear();
}

/*
```

Comportamiento cuando se presiona el botón Seleccionar.

```
*/  
function evt__cuadro__seleccion($datos)  
{  
    $this->dep('datos')->cargar($datos);  
    $this->set_pantalla('pant_edicion');  
}
```

## Consola de comandos

El objetivo de estos comandos es administrar una instalación completa de toba. Teniendo en cuenta que una instalación se divide en instancias y éstas a su vez en proyectos, los principales comandos representan a cada uno de estos conceptos.

### **Instalación**

- Crear completamente con una instancia y un proyecto inicial.
  - toba instalación autoinstalar
- Crear solamente la instalación:
  - toba instalación crear

### **Instancia**

Cuando se quiere trabajar con una instancia específica, el identificador de ésta debe pasarse como parámetro en cada invocación de comando (eg. -i id\_instancia), sino debe incluirse en la variable de entorno toba\_instancia (eg. toba\_instancia=desarrollo).

- Crear una nueva instancia. Pregunta por los proyectos a incluir y la base de datos a utilizar en esta nueva instancia.
  - toba instancia crear
- Vuelve a generar la instancia a partir de los metadatos existentes en el sistema de archivos. Previamente, en forma opcional, exporta la información local:
  - toba instancia regenerar
- Exporta los metadatos locales de la instancia (usuarios, logs, etc.) que no forman parte de ningún proyecto particular:
  - toba instancia exportar\_local

### **Proyecto**

Cuando se quiere trabajar con un proyecto específico, el identificador de éste debe pasarse como parámetro en cada invocación de comando (eg. -p id\_proyecto), sino debe incluirse en la variable de

## SIU-TOBA - Ambiente de Desarrollo Web

entorno toba\_proyecto (eg. toba\_proyecto=contratos).

- Crear un nuevo proyecto y lo agrega a la instancia actual. Opcionalmente crea también el alias de apache:
  - toba proyecto crear
- Exporta los metadatos del proyecto, y por seguridad exporta la información local de la instancia también (toba instancia exportar\_local):
  - toba proyecto exportar
- Regenera únicamente los metadatos del proyecto, dejando intacta el resto de la instancia:
  - toba proyecto regenerar

## Referencias:

- Descargar TOBA:  
<http://toba.siu.edu.ar/trac/toba>
- Como instalar TOBA (Linux y Windows):  
<http://toba.siu.edu.ar/trac/toba/wiki/Instalacion>
- Configuración Postgres:  
<http://www.ubuntugeek.com/howto-setup-database-server-with-postgresql-and-pgadmin3.html>