

# DISEÑO DE APLICACIONES



Alejandro Tocar



[aletocar@gmail.com](mailto:aletocar@gmail.com)



[aletocar](https://github.com/aletocar)

# TEMAS

1. Introducción a ASP.NET Web API 2
2. Ventajas de ASP.NET Web API
3. Características de ASP.NET Web API
4. Introducción a REST
5. Comienzo del desarrollo de Tresana

# **INTRODUCCIÓN A ASP.NET WEB API 2**

# ASP.NET COMO UN FRAMEWORK DE SERVICIOS

1. Presentación .NET - ASMX y SOAP
2. .NET 1.0 - Remoting
3. .NET 3.0 - WCF

# **JAVASCRIPT Y MOBILE**

# **VENTAJAS DE WEB API**

# CONFIGURACIÓN

# REST POR DEFECTO

```
public class TasksController : ApiController
{
    private readonly IRepository repository;

    public TasksController(IRepository repository)
    {
        repository = repository;
    }

    public Task Get(long taskId)
    {
        return Json(_repository.Get(taskId));
    }
}
```



# ABSTRACCIONES CON RUTAS

# CONTROLLER ACTIVATION

# **INTEROPERABILIDAD CON JSON, XML Y REST**

# CARACTERÍSTICAS

1. CRUD por convención
2. Negociación de contenido
3. Ruteo por atributo
4. Restricciones en rutas
5. Soporte de CORS
6. Manejo global de errores
7. IHttpActionResult

# INTRODUCCIÓN A REST

# ¿QUÉ ES RESTFUL?

1. URIs y Recursos
2. Verbos HTTP
3. Códigos HTTP

# URIS Y RECURSOS

# VERBOS HTTP

Método	URI	Verbo HTTP	Contrato/Cambio de estado
CreateTask	/api/tasks	POST	HTTP POST usado para creación de Tarea
GetTask	/api/tasks/1234	GET	HTTP GET usado para obtener una Tarea
GetTaskAsignees	/api/tasks/1234/users	GET	HTTP GET usado para obtener los usuarios asignados a una tarea
SearchTasks	/api/tasks	GET	HTTP GET usado para obtener una lista de tareas
UpdateTask	/api/tasks/1234	PUT	HTTP PUT usado para modificar la Tarea
CompleteTask	/api/tasks/1234	DELETE	HTTP DELETE usado para eliminar la Tarea



# CÓDIGOS DE ESTADO HTTP

Status Code	Significado
200	All is good; response will include applicable resource information, as well
201	Resource created; will include the Location header specifying a URI to the newly created resource
202	Same as 200, but used for async; in other words, all is good, but we need to poll the service to find out when completed
301	The resource was moved; should include URI to new location
400	Bad request; caller should reformat the request
401	Unauthorized; should respond with an authentication challenge, to let the caller resubmit with appropriate credentials
402	Reserved for future use, but many times used to indicate a failure in a business rule or validation check
403	Access denied; user successfully authenticated, but is not allowed to access the requested resource
404	Resource not found, or caller not allowed to access the resource, and we don't want to reveal the reason
409	Conflict; used as a response to a PUT request when another caller has dirtied the resource
500	Server error; something bad happened, and server might include some indication of the underlying problem