

# Desarrollo de juegos en HTML5

*Nicolás Mendoza R.*

# Ambiente de desarrollo

- CAAT (Canvas Advanced Animation Toolkit)
- IDE recomendado: Aptana Studio 3



aptana<sup>®</sup>

# Instrucciones de instalación

- Bajar la versión "Standalone" (Recomendado)
- Abrir el wizard de instalación y elegir las preferencias

## Download Aptana Studio 3.2.2

Aptana Studio 3.2.2 is our code base and complete environment that includes extensive capabilities to build Ruby and Rails, PHP, and Python applications, along with complete HTML, CSS and JavaScript editing.

You are about to download:

**Aptana Studio 3.2.2**  
Microsoft Windows, x86 (x64 Compatible)

Standalone Version  Eclipse Plug-in Version

**System Requirements**  
Windows XP or newer

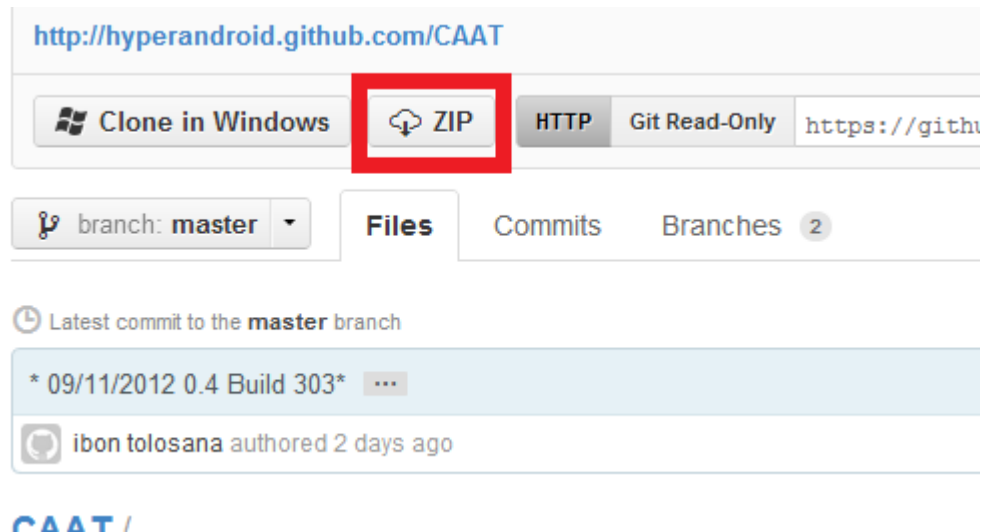
[Getting Started Guide](#)

[Release Notes](#)

[Looking for a different OS, download format, or architecture?](#) [Customize Your Download](#)

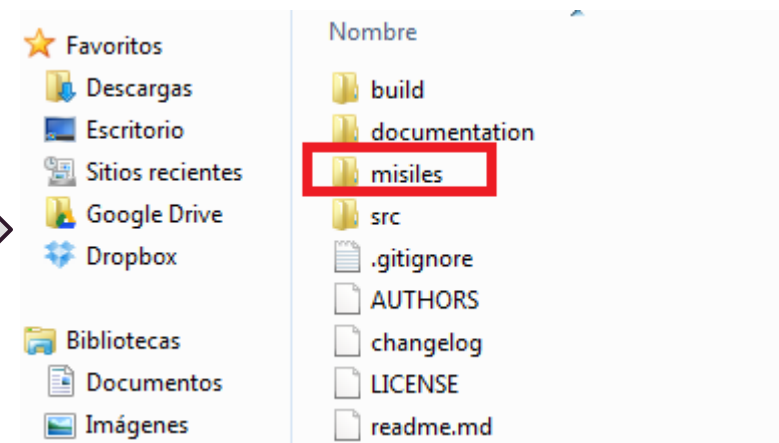
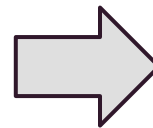
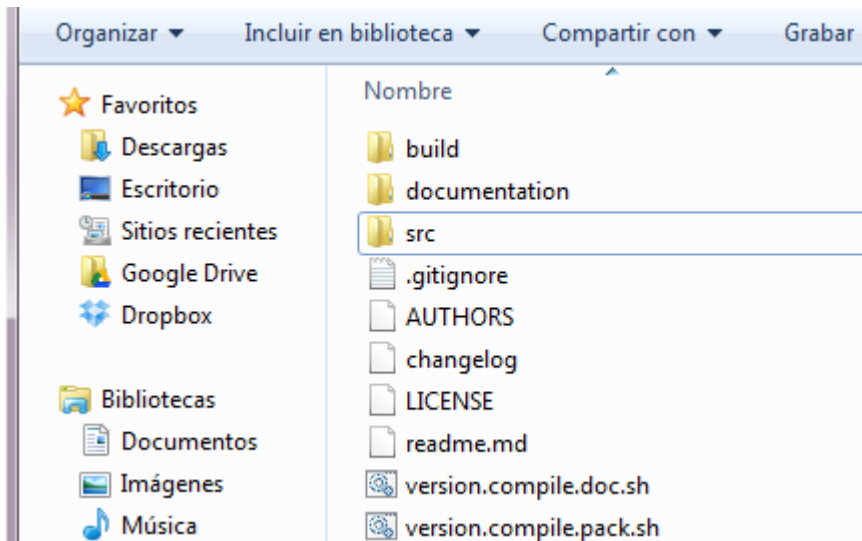
# Instrucciones de instalación

- Bajar el código fuente de github
- Descomprimir el archivo



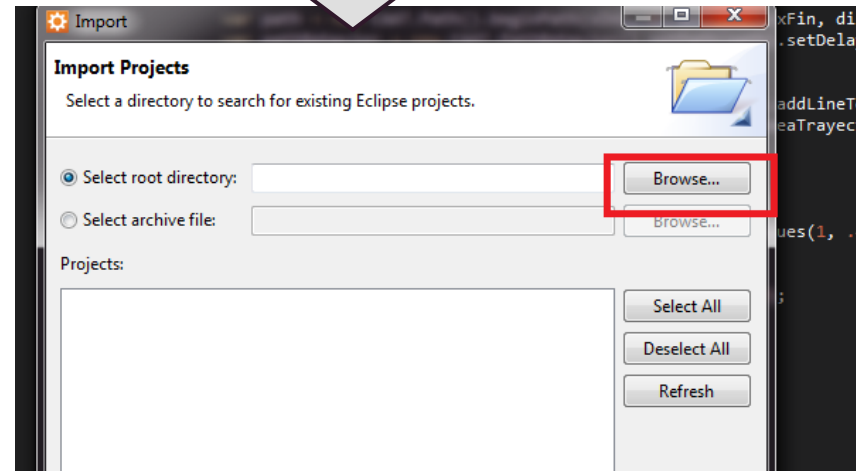
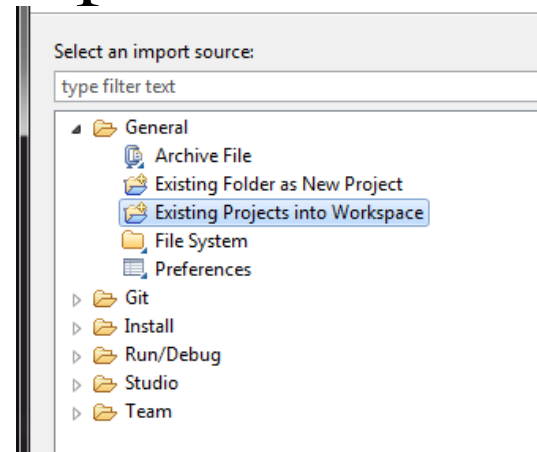
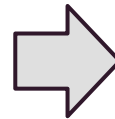
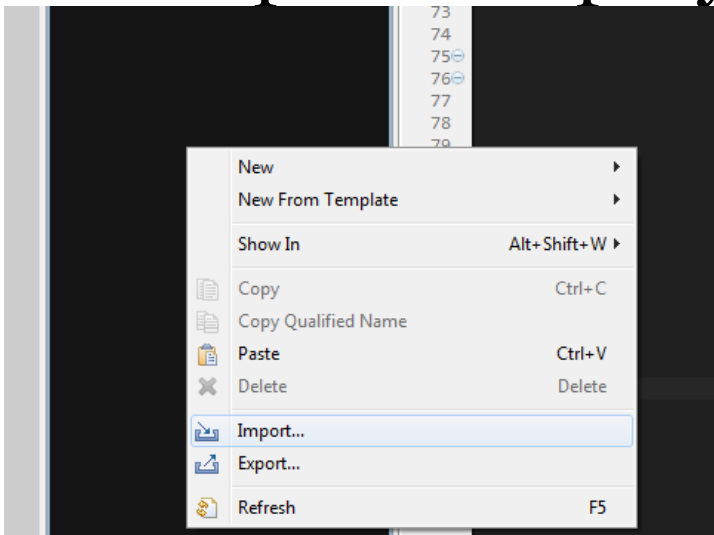
# Instrucciones de instalación

- Agregar la carpeta del juego en la carpeta base de CAAT.



# Instrucciones de instalación

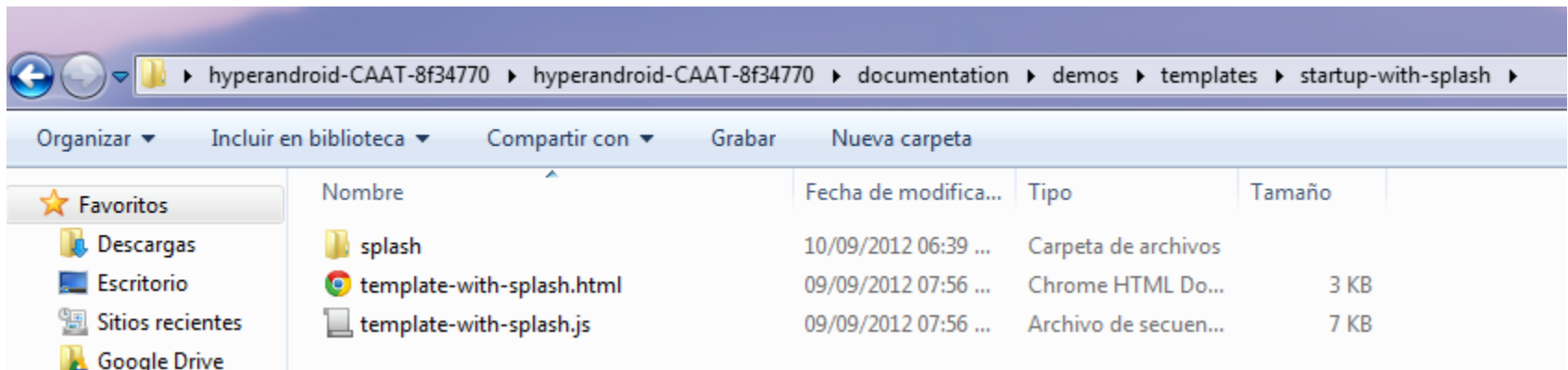
- Importar el proyecto a Aptana



# Plantilla inicial

- Ruta para el plantilla inicial:

- <Carpeta del proyecto>/documentation/demos/templates/startup-with-splash



# Punto de inicio

- El código en el cuadro se puede borrar para dar inicio al juego que vamos a hacer.

```
/**
 * This function will be called to let you define new scenes that will be
 * shown after the splash screen.
 * In this example, create a simple scene with some circles on it.
 * @param director {CAAT.Director}
 */
function createScenes(director) { Punto de inicio del juego
  var scene = director.createScene();

  for (var i = 0; i < 30; i++) {
    var w = 30 + (100 * Math.random()) >> 0;
    var r = (255 * Math.random()) >> 0;
    var g = (255 * Math.random()) >> 0;
    var b = (255 * Math.random()) >> 0;
    scene.addChild(
      new CAAT.Actor().
        setBounds(
          (director.width * Math.random()) >> 0,
          (director.height * Math.random()) >> 0,
          w,
          w),
      setFillStyle('rgb(' + r + ',' + g + ',' + b + ')')
    );
  }
};

/**
 * Startup it all up when the document is ready.
 * Change for your favorite frameworks initialization code.
 */
window.addEventListener(
  'load',
  function() {
```

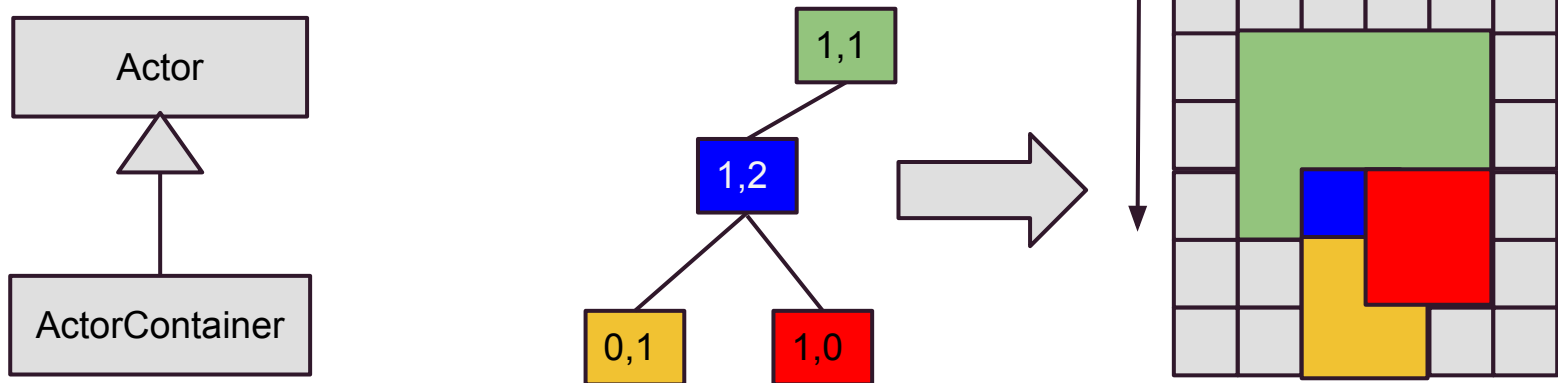


# Clases principales

- **Director**
  - Objeto principal del framework. Encargado de la renderización y el manejo de escenas.
- **Scene**
  - Escena del juego. Contiene todos los objetos visibles dentro del juego. Se puede pensar como un nivel dentro del juego.
- **Actor**
  - Objeto visible dentro de la escena. Es un cuadrado al cual se le puede asignar un color (o una imagen), cambiar tamaños, animar, etc.

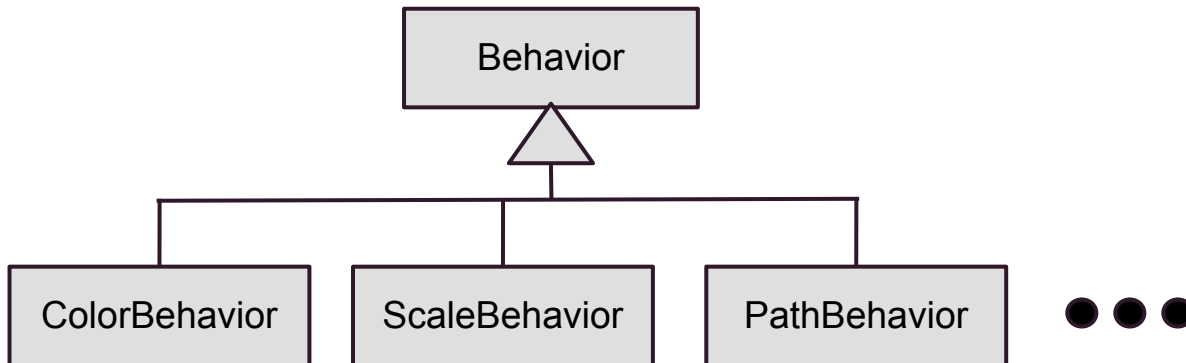
# Clases principales

- ActorContainer
  - Hereda de Actor. Es un Actor el cual puede contener varios actores dentro de él. Las transformaciones aplicadas a esta clase, también se aplicarán a los actores que contenga.



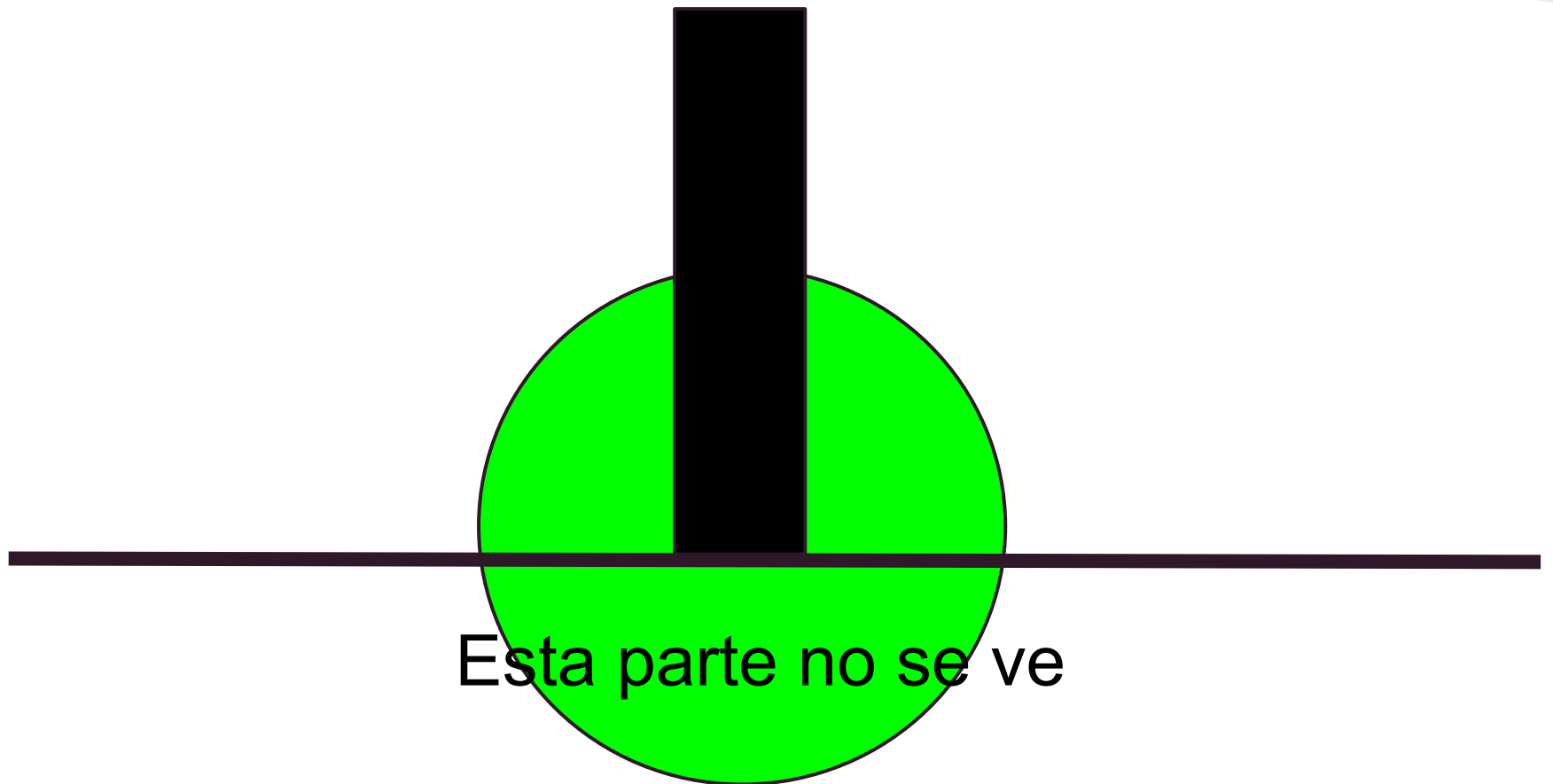
# Clases principales

- Behavior
  - Clase que expone servicios para aplicar transformaciones dinámicas (animaciones) a los actores. Otras clases heredan de esta clase para ofrecer dichos servicios.



# Composición de la torreta

La torreta es un círculo con un rectángulo.



# Composición de los misiles

- Un misil se compone de un cuadrado y un círculo.

