

# **Desarrollo de aplicaciones para móviles**

**Centro de e-Learning SCEU UTN - BA.**

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

**[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)**

## **Modulo 1: Programación y plataformas**

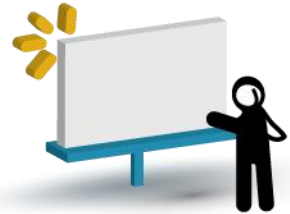
### **Unidad 2: HTML 5 - Ajax**

---



**UTN.BA**  
UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES

**Centro de  
e-Learning**



## Presentación:

En esta unidad vemos lo nuevo en HTML, el estándar HTML 5. También vamos a conocer sus principales funcionalidades y ventajas. Debemos recordar que IONIC (el framework elegido para el desarrollo de aplicaciones móviles) está basado en este estándar.

Por otro lado, incursionamos en la utilización de AJAX (Asynchronous JavaScript And XML), el cual es una técnica de programación web para crear aplicaciones RIA (Rich Internet Applications).

**Centro de e-Learning SCEU UTN - BA.**

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

**[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)**



## Objetivos:

### Que los participantes:

- Logren un conocimiento avanzado en el estándar HTML 5
- Entiendan cuáles son los principales recursos de HTML 5 en la futura aplicación para aplicaciones móviles.
- Incorporen HTML 5 para facilitar la utilización de recursos como videos y audios
- Entiendan el funcionamiento de AJAX para su utilización en unidades posteriores.



## Bloques temáticos:

HTML 5.

Cuáles son las novedades de HTML 5.

Semántica.

Almacenamiento.

Multimedia.

UTN Viajes.

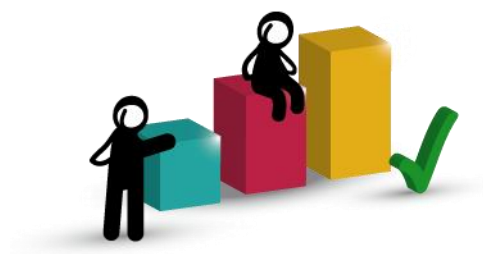
Ajax.

Un ejemplo de utilización.

Ajax request.

Ajax response.

Sigamos de viaje...



## Consignas para el aprendizaje colaborativo

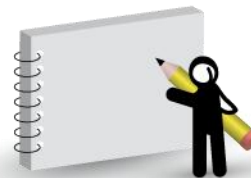
En esta Unidad los participantes se encontrarán con diferentes tipos de actividades que, en el marco de los fundamentos del MEC\*, los referenciarán a tres comunidades de aprendizaje, que pondremos en funcionamiento en esta instancia de formación, a los efectos de aprovecharlas pedagógicamente:

- Los foros proactivos asociados a cada una de las unidades.
- La Web 2.0.
- Los contextos de desempeño de los participantes.

Es importante que todos los participantes realicen algunas de las actividades sugeridas y compartan en los foros los resultados obtenidos.

Además, también se propondrán reflexiones, notas especiales y vinculaciones a bibliografía y sitios web.

El carácter constructivista y colaborativo del MEC nos exige que todas las actividades realizadas por los participantes sean compartidas en los foros.



## Tomen nota:

Las actividades son opcionales y pueden realizarse en forma individual, pero siempre es deseable que se las realice en equipo, con la finalidad de estimular y favorecer el trabajo colaborativo y el aprendizaje entre pares. Tenga en cuenta que, si bien las actividades son opcionales, su realización es de vital importancia para el logro de los objetivos de aprendizaje de esta instancia de formación. Si su tiempo no le permite realizar todas las actividades, por lo menos realice alguna, es fundamental que lo haga. Si cada uno de los participantes realiza alguna, el foro, que es una instancia clave en este tipo de cursos, tendrá una actividad muy enriquecedora.

Asimismo, también tengan en cuenta cuando trabajen en la Web, que en ella hay de todo, cosas excelentes, muy buenas, buenas, regulares, malas y muy malas. Por eso, es necesario aplicar filtros críticos para que las investigaciones y búsquedas se encaminen a la excelencia. Si tienen dudas con alguno de los datos recolectados, no dejen de consultar al profesor-tutor. También aprovechen en el foro proactivo las opiniones de sus compañeros de curso y colegas.

## HTML 5

---

Como se menciona en la unidad anterior HTML 5 no es simplemente una nueva versión del lenguaje de marcación HTML, sino una agrupación de diversas especificaciones concernientes al desarrollo web. Es decir, HTML 5 no se limita sólo a crear nuevas etiquetas, atributos y eliminar aquellas marcas que están en desuso o se utilizan inadecuadamente, sino que va mucho más allá.



Así pues, HTML 5 es una nueva versión de diversas especificaciones, entre las que se encuentran:

- HTML 4
- XHTML 1
- CSS Nivel 2
- DOM Nivel 2 (DOM = Document Object Model)

A la par, HTML 5 pretende proporcionar una plataforma con la que desarrollar aplicaciones web más parecidas a las aplicaciones de escritorio, donde su ejecución dentro de un navegador no implique falta de recursos o facilidades para resolver las necesidades reales de los desarrolladores.



## **Cuáles son las novedades de HTML 5**

---

HTML 5 incluye novedades significativas en diversos ámbitos. Este nuevo estándar supone mejoras en áreas que hasta ahora quedaban fuera del lenguaje y para las que se necesitaba utilizar otras tecnologías.

- Estructura del cuerpo: La mayoría de las webs tienen un formato común, formado por elementos como cabecera, pie, navegadores, etc. HTML 5 permite agrupar todas estas partes de una web en nuevas etiquetas que representarán cada uno de las partes típicas de una página.
- Etiquetas para contenido específico: Hasta ahora se utilizaba una única etiqueta para incorporar diversos tipos de contenido enriquecido, como animaciones Flash o vídeo. Ahora se utilizarán etiquetas específicas para cada tipo de contenido en particular, como audio, vídeo, etc.
- Canvas: es un nuevo componente que permitirá dibujar, por medio de las funciones de un API, en la página todo tipo de formas, que podrán estar animadas y responder a interacción del usuario. Es algo así como las posibilidades que nos ofrece Flash, pero dentro de la especificación del HTML y sin la necesidad de tener instalado ningún plugin.
- Bases de datos locales: el navegador permitirá el uso de una base de datos local, con la que se podrá trabajar en una página web por medio del cliente y a través de un API. Es algo así como las Cookies, pero pensadas para almacenar grandes cantidades de información, lo que permitirá la creación de aplicaciones web que funcionen sin necesidad de estar conectados a Internet.

- Web Workers: son procesos que requieren bastante tiempo de procesamiento por parte del navegador, pero que se podrán realizar en un segundo plano, para que el usuario no tenga que esperar que se terminen para empezar a usar la página.
- Aplicaciones web Offline: Existirá otro API para el trabajo con aplicaciones web, que se podrán desarrollar de modo que funcionen también en local y sin estar conectados a Internet.
- Geolocalización: Las páginas web se podrán localizar geográficamente por medio de un API que permita la Geolocalización.
- Nuevas APIs para interfaz de usuario: temas tan utilizados como el "drag & drop" (arrastrar y soltar) en las interfaces de usuario de los programas convencionales, serán incorporadas al HTML 5 por medio de un API.
- Fin de las etiquetas de presentación: todas las etiquetas que tienen que ver con la presentación del documento, es decir, que modifican estilos de la página, serán eliminadas. La responsabilidad de definir el aspecto de una web correrá a cargo únicamente de CSS.

## Semántica

---

### Secciones y contenido en HTML 5



Todo el contenido incluido dentro del elemento `<body>` es parte de una sección. Las secciones en HTML5 pueden ser anidadas. Además de la sección principal, definida por el elemento `<body>`, los límites de la sección son definidos explícita o implícitamente. Las secciones definidas explícitamente son el contenido definido en las etiquetas `<body>`, `<section>`, `<article>`, `<aside>`, `<footer>`, `<header>`, y `<nav>`.

Por ejemplo:

```
1  <section>
2    <h1>El pato</h1>
3    <section>
4      <h1>Introducción</h1>
5      <p>En esta sección, ampliaremos nuestro concepto del pato.
6    </section>
7    <section>
8      <h1>Hábitat</h1>
9      <p>El pato, como fiero depredador, necesita un entorno con abundantes lobos que cazar.
10   </section>
11   <aside>
12     <p>otros estudiosos del pato
13   </aside>
14 </section>
15 <footer>
16   <p>(c) 2010 The Example company
17 </footer>
```

El cual genera la siguiente salida:

```

1 | 1. El pato
2 |   1.1 Introducción
3 |   1.2 Hábitat
4 |   1.3 Section (aside)

```

## Seccionado implícito

Los elementos de cabecera (<h1> a <h6>) definen un nuevo seccionado implícito cuando ellos no son el primer encabezado de sus secciones padre. La forma en que esta sección implícita es posicionada en el esquema es definida por su rango relativo con la cabecera anterior en su sección padre. Si es de un rango más bajo que la cabecera anterior, abre una sub-sección implícita de la sección.

Por ejemplo:

```

1 | <section>
2 |   <h1>El águila</h1>
3 |   <p> ....
4 |
5 |   <h3 class="subsec-implicita">Hábitat</h3>
6 |   <p> ...
7 | </section>

```

Genera la siguiente salida:

```

1 | 1. El águila
2 |   1.1 Hábitat (impícitamente definido mediante <h3>)

```

Si es del mismo rango que la cabecera anterior, cierra la sección previa y abre una nueva sección implícita del mismo nivel:

```
1 <section>
2   <h1>El águila</h1>
3   <p>...
4   <h1 class="secc-implicita">El buitre</h1>
5   <p>...
6 </section>
```

Genera el siguiente esquema:

```
1 1. El águila
2 2. El buitre (implícitamente definido por <h1>, quien al mismo tiempo cierra el <h1> anterior)
```

## Article

El Elemento article de HTML (<article>) representa una composición auto-contenida en un documento, página, una aplicación o en el sitio, que se destina a distribuir de forma independiente o reutilizable. Podría ser un mensaje en un foro, un artículo de una revista o un periódico, una entrada de blog, un comentario de un usuario, un widget interactivo o gadget, o cualquier otro elemento independiente del contenido.

Por ejemplo Cuando los elementos <article> están anidados, los internos representan artículos relacionados con el exterior. Por ejemplo, los comentarios de un blog pueden ser elementos <article> anidados al que representa la entrada del blog.

## Nav

Utilizado para crear un menú de navegación

```
<nav>

  <ul>

    <li><a href="DireccionPagina"> Item de Navegación 1 </li>

    <li><a href="DireccionPagina"> Item de Navegación 2 </li>

    <li><a href="DireccionPagina"> Item de Navegación Etc </li>

  </ul>

</nav>
```

## Aside

El Elemento HTML Aside (<aside>) representa una sección de una página que consiste en contenido que podría ser considerado independiente del contenido que lo rodea. Estas secciones son a menudo representadas como barras laterales o como inserciones y contienen una explicación al margen como una definición de glosario, elementos relacionados indirectamente, como publicidad, la biografía del autor, o en aplicaciones web, la información de perfil o enlaces a blogs relacionados.

## Formularios en HTML 5

### Input

El elemento <input> tiene nuevos valores para el atributo type.

- search: El atributo representa un campo de búsqueda. Los saltos de línea son automáticamente eliminados del valor.
- tel: El atributo representa un control para editar un número de teléfono. Los saltos de línea son automáticamente eliminados, ya que los números telefónicos varían

enormemente en todo el mundo. Puedes usar atributos como `pattern` y `maxlength` para restringir los valores ingresados en el campo.

- `url`: El atributo representa un control para editar una URL. Se eliminan los saltos de línea y espacios en blanco antes y después del valor ingresado.
- `email`: El atributo representa una dirección de correo electrónico. Los saltos de línea se eliminan automáticamente del valor ingresado.

El elemento `<input>` también tiene nuevos atributos:

- `pattern`: una expresión regular respecto a la cual se comprueba el valor del control, que puede ser usada con valores de `type` de `text`, `tel`, `search`, `url` y `email`.

## Output

El elemento `<output>` representa el resultado de un cálculo.

Puedes usar el atributo `for` para especificar una relación entre el elemento `output` y otros elementos en el documento que afectan el cálculo (por ejemplo, ingreso de datos o parámetros). El valor del atributo `for` es una lista separada por espacios de identificadores de otros elementos.

## Atributo placeholder

El atributo `placeholder` en elementos `<input>` y `<textarea>` proporciona una ayuda a los usuarios acerca de qué debe ingresarse en el campo. El texto introducido en el `placeholder` no debe contener retornos de carro o saltos de línea.

## Validación restringida

Los siguientes elementos de sintaxis de HTML5 pueden ser usados para restringir datos en el formulario:

- El atributo `required` en los elementos `<input>`, `<select>` y `<textarea>` indica que se debe ingresar algún dato. (En el elemento `<input>`, `required` solo se aplica con ciertos valores del atributo `type`.)
- El atributo `pattern` en el elemento `<input>` restringe el valor para que concuerde con una expresión regular específica.
- Los atributos `min` y `max` del elemento `<input>` restringen los valores máximos y mínimos que pueden ser ingresados.
- El atributo `step` del elemento `<input>` (cuando se usa en combinación con los atributos `min` y `max`) restringe la granularidad de los valores ingresados. Un valor que no se corresponda con un valor permitido no será validado.
- El atributo `maxlength` de los elementos `<input>` y `<textarea>` restringe el máximo número de caracteres (en puntos de código unicode) que el usuario puede ingresar.
- Los valores `url` y `email` para `type` restringen el valor para una URL o dirección de correo válida respectivamente.

Además, puedes evitar la validación restringida especificando el atributo `novalidate` en el elemento `<form>`, o el atributo `formnovalidate` en el elemento `<button>` y en el elemento `<input>` (cuando `type` es `submit` o `image`). Estos atributos indican que el formulario no será validado cuando se envíe.



## Almacenamiento

---

HTML 5 nos provee de persistencia a través del navegador, a continuación veremos el elemento más utilizado para lograr la misma



### Base de datos en HTML 5

Aunque LocalStorage nos abre un mundo nuevo de posibilidades, no deja de tener ciertas limitaciones como el espacio (normalmente unos 10MB en total) o la imposibilidad de hacer búsquedas... y aquí entra en juego IndexedDB.

Cuando hablamos de base de datos, nos vienen a la mente las bases de datos relacionales y en el SQL. En la elaboración del estándar HTML5 este detalle se tuvo en cuenta con el WebSQL. WebSQL era una de las novedades más increíbles de HTML5, puesto que permitía atacar una base de datos SQLite con la API de Javascript que HTML5 nos proporcionaba para ello.

### Indexed BD

IndexedDB es la herramienta que HTML5 proporciona a los programadores para poder trabajar con una base de datos desde la propia aplicación web, sin tener que conectarnos a ningún servidor... disponible offline (en entornos sin conexión web) y persistente (al cerrar la aplicación, el contenido no se borra).

Como algunas de las bondades de IndexedDB podemos destacar las siguientes:

- **Es un sistema de almacenamiento local:** Es una característica muy útil; pensemos en una aplicación móvil hecha en HTML o en una extensión de Chrome, Firefox, etc. Si IndexedDB no fuera un sistema de almacenamiento local no podríamos acceder al contenido almacenado si nuestra conexión de red estuviese caída. Gracias a que la información almacenada en IndexedDB está en el propio equipo, nuestra aplicación puede seguir funcionando perfectamente aunque estemos sin conexión.
- **No tiene límite de tamaño:** Es una de sus grandes bondades. Al usar LocalStorage, tenemos un límite de almacenamiento establecido por el navegador... límite que suele rondar los 10MB. Si bien es suficiente para gran parte de los casos, este límite puede suponer un problema si nuestra aplicación mueve mucho dato. En el caso de IndexedDB este límite no existe y estamos hablando de un sistema de almacenamiento ilimitado (bueno, limitado por tu disco duro claro). Eso sí, en el caso de Firefox, aunque el espacio sea ilimitado, el navegador pre-reserva 50MB de espacio y te solicita permiso cuando necesita reservar otros 50MB.
- **Permite múltiples “tablas”:** Bueno, IndexedDB no funciona con tablas, pero para que nos entendamos, puedes tener en una misma base de datos varias tablas para almacenar tus contenidos.
- **Permite múltiples bases de datos:** No estás limitado a una base de datos, puedes conectarte a varias bases de datos vía IndexedDB.

Veremos más de indexed DB cuando veamos la unidad 11 (trabajar offline) por ahora alcanza con que recuerdes que es una base de datos.

## Multimedia

---



HTML 5 nos provee de ciertas etiquetas para facilitar la introducción de elementos de audio y video

### Video

```
1 <video src="http://v2v.cc/~j/theora_testsuite/320x240.ogg" controls>
2   Tu navegador no implementa el elemento <code>video</code>.
3 </video>
```

Este ejemplo reproduce un vídeo de ejemplo, con los controles de reproducción, desde el sitio Web de Theora.

Se pueden especificar múltiples fuentes de archivos usando el elemento `<source>` con el fin de proporcionar vídeo o audio codificados en formatos diferentes para diferentes navegadores.

Por ejemplo:

```
1 <video controls>
2   <source src="foo.ogg" type="video/ogg">
3   <source src="foo.mp4" type="video/mp4">
4   Tu navegador no implementa el elemento <code>video</code>.
5 </video>
```

## Audio

```
1 <audio src="/test/audio.ogg">
2   <p>Tu navegador no implementa el elemento audio.</p>
3 </audio>
```

Algunos atributos del elemento audio:

- controls : muestra los controles estándar de HTML5 para audio en una página web.
- autoplay : hace que el audio se reproduzca automáticamente.
- loop : hace que el audio se repita automáticamente.

El atributo preload es usado en el elemento audio para almacenar temporalmente (buffering) archivos de gran tamaño. Este puede tomar uno de 3 valores:

- "none" no almacena temporalmente el archivo
- "auto" almacena temporalmente el archivo multimedia
- "metadata" almacena temporalmente sólo los metadatos del archivo

## Controlando la reproducción multimedia

Mediante la utilización de javascript podemos controlar la reproducción de los elementos multimedia insertados, por ejemplo mediante la función **play()** iniciaremos la reproducción de un audio o video.

Más ejemplos:

```
1 <audio id="demo" src="audio.mp3"></audio>
2 <div>
3   <button onclick="document.getElementById('demo').play()">Reproducir el Audio</button>
4   <button onclick="document.getElementById('demo').pause()">Pausar el Audio</button>
5   <button onclick="document.getElementById('demo').volume+=0.1">Aumentar el Volumen</button>
6   <button onclick="document.getElementById('demo').volume-=0.1">Disminuir el Volumen</button>
7 </div>
```

Podes encontrar más información acerca de este tema en:  
[https://developer.mozilla.org/es/docs/Web/HTML/Usando\\_audio\\_y\\_video\\_con\\_HTML5](https://developer.mozilla.org/es/docs/Web/HTML/Usando_audio_y_video_con_HTML5)

## UTN Viajes

---

La facultad regional Buenos Aires de la Universidad Tecnológica Nacional está impulsando viajes por la República Argentina para sus estudiantes.

En una primera etapa la UTN tiene habilitado por la secretaria de turismo de la nación los siguientes centros turísticos:

- Córdoba|
- Puerto Madryn
- Salta capital

Para poder difundir estos viajes la facultad ha decidido contratar a una prestigiosa agencia de diseño a la cual se le encargo el desarrollo de las respectivas landing pages de estos centros turísticos.

Cada landing page es una página “independiente” que puede mantener ciertos criterios en el diseño.

Cada página debe contener información acerca del viaje y un video que nos muestre los paisajes del lugar. También se deja a criterio del diseñador el agregado de más información.

## Ajax

---



AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.



## Un ejemplo de utilización

---

```
<!DOCTYPE html>
<html>
<body>

<div id="demo"><h2>Enviar contacto</h2></div>

<div>
  <label>Nombre</label>
  <input name="nombre" id="nombre"/>
</div>

<button type="button" onclick="enviar_formulario()">Change Content</button>

</body>
</html>

<script type="text/javascript">
  function enviar_formulario() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
      if (xhttp.readyState == 4 && xhttp.status == 200) {
        document.getElementById("demo").innerHTML = xhttp.responseText;
      }
    };
    xhttp.open("GET", "ajax_info.txt", true);
    xhttp.send();
  }
</script>
```

Al clicar en el boton se ejecutara la petición ajax que nos traerá la información del archivo ajax\_info.txt





## Ajax request

---

```
xhttp.open("GET", "ajax_info.txt", true);  
xhttp.send();
```

### Open

Parámetro 1: Tipo de request

- GET
- POST: Este tipo de request se utiliza cuando la cantidad de datos a enviar en la petición ajax es muy extensa. POST es más robusto y seguro que GET.

Parámetro 2: Url del archivo

Parámetro 3: Asíncrono -> true o false. En caso de ser false se ejecutara de forma sincrónica, es decir se ejecutara la petición ajax y no se ejecutara el resto de código hasta que la misma no sea respondida.

### Send

Con este método enviamos el request ajax al servidor.



## Ajax response

---

Una vez realizada la petición ajax al servidor esperamos una respuesta (response) del mismo para poder trabajar con los datos devueltos.

### responseText

```
document.getElementById("demo").innerHTML = xhttp.responseText;
```

Este método es utilizado cuando la respuesta del servidor no es un XML sino que es un texto (string).

### responseXML

En caso de que la respuesta del servidor no sea un texto (string) y sea un XML, entonces debemos “parsear” dicha respuesta.

```
xmlDoc = xhttp.responseXML;  
txt = "";  
x = xmlDoc.getElementsByTagName("ARTIST");  
for (i = 0; i < x.length; i++) {  
    txt += x[i].childNodes[0].nodeValue + "<br>";  
}  
document.getElementById("demo").innerHTML = txt;
```

Para mas información podés visitar la siguiente página <http://www.w3schools.com/ajax/>

## Sigamos de viaje...

---

Vamos a mejorar nuestra landing page de UTN viajes. Se nos solicita un nuevo requerimiento, el mismo consiste en que cada página tenga un formulario de contacto para que los estudiantes puedan contactarse para averiguar más información acerca de los viajes.

En esta etapa del desarrollo no se plantea la persistencia de los datos (es decir el guardado de los mismos).

El formulario de contacto debe contener los siguientes campos:

- Nombre
- Apellido
- Email
- Asunto
- Mensaje

El desarrollo que debemos realizar es que al completar el formulario y apretar enviar se debe mostrar un mensaje de agradecimiento el cual debe estar guardado en un archivo txt externo.



**UTN.BA**  
UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES

**Centro de  
e-Learning**



## Bibliografía utilizada y sugerida

<https://es.wikipedia.org/wiki/HTML>

<https://developer.mozilla.org/es/docs/HTML/HTML5>

<https://es.wikipedia.org/wiki/AJAX>

<http://www.w3schools.com/ajax/>

**Centro de e-Learning SCEU UTN - BA.**

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

**[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)**

## Lo que vimos:

En esta unidad hemos aprendido más acerca de lo nuevo que nos trajo el estándar de HTML 5. Por otro lado hemos tenido un simple vistazo acerca de ajax, el cual nos permitirá entender ciertos funcionamientos que veremos en la unidad de angular JS.



---

## Lo que viene:

En la próxima unidad entraremos en un nuevo mundo... Angular JS.

Prepárate y junta energías porque el próximo desafío es aun más importante.

Te esperamos para afrontarlos juntos!!!

