



# Astro

Framework para la creación de sitios web

<https://astro.build/>



Juan Vladimir  
@juanvladimir13

# Agenda

1. Que es ?
2. Creación del entorno de desarrollo
3. Comandos de bun.sh
4. Estructura de un proyecto
5. "Hola mundo"
6. Renderizado de variables y objetos
7. Creación y uso de componentes
8. Layouts
9. Renderizado de un array de objetos

# Que es Astro

**Astro** es el framework web para construir sitios web que cargue rápidamente y tenga un excelente SEO, webs orientados al contenido, como blogs, marketing y comercio electrónico.



# Creación del entorno de desarrollo

## 1. Instalación del compilador “bun”

Ejecutar el **powershell** e introducir el siguiente comando

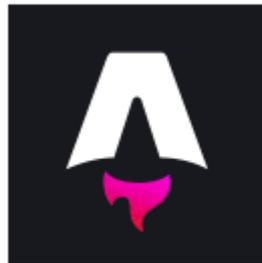
```
powershell -c "irm bun.sh/install.ps1 | iex"
```

## 2. Instalación del plugin “astro” en VSCode

Ir al menu **View** → **Extensions**

Buscar la extension con el nombre de **astro**

Instalar el plugin



### Astro

Astro  astro.build |  989,437 |  (35)

Language support for Astro

Disable 

Uninstall 

Auto Update 

# Comandos bun para proyectos de astro

```
bun create astro
```

*Inicia un wizard para crear un proyecto*

```
bun run dev
```

*Inicia el servidor de desarrollo local, si todo sale bien, ahora el proyecto esta en <http://localhost:4321>*

```
bun run build
```

*Compila una versión del sitio web lista para ser desplegada en una carpeta separada llamada **dist***

# Estructura de un proyecto

**src/\*** - Código fuente del proyecto (componentes, páginas, estilos, imágenes, etc.)

**public/\*** - Archivos sin código que no necesitan ser procesados durante la compilación como imágenes y fuentes o archivos especiales

**package.json** - El manifiesto de tu proyecto

**astro.config.mjs** - El archivo de configuración de Astro

**tsconfig.json** - El archivo de configuración de TypeScript



# Estructura del directorio src

## **src/pages**

Archivos de páginas web que se crean

## **src/components**

Archivos reutilizables de código HTML para las paginas web.

## **src/layouts**

Son **componentes** que definen la estructura de la UI compartida por una o más páginas.

## **src/styles**

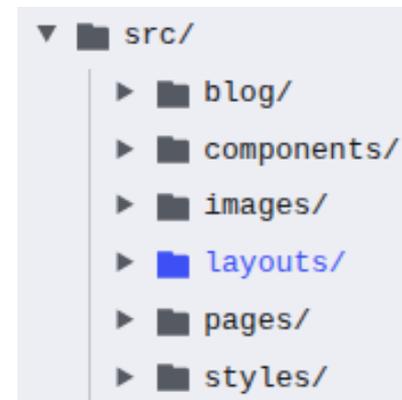
Directorio para guardar archivos CSS.

## **src/contents**

Directorio para guardar archivos con contenido o data, archivos permitidos son \*.json, \*.yaml, etc.

## **src/scripts**

Directorio para guardar archivos de tipo \*.js o \*.ts que manipulan archivos \*.html



# “Hola mundo”

Borrar todo el contenido del archivo **src/pages/index.astro**

Escribir las siguientes líneas de código.

---

---

```
<h1>Hola mundo</h1>
```

# Renderizado de variables y objetos

Borrar todo el contenido del archivo `src/pages/index.astro`

Escribir las siguientes líneas de código.

----

```
const welcomeGreeting = "Hola mundo";
const soccerPlayer = {
  name: "Cristiano Ronaldo",
  nick: "CR7",
  nro: 7
};
```

----

```
<h1>Mi nuevo sitio Astro</h1>
<h2> {welcomeGreeting} </h2>
<span> {soccerPlayer.nro} </span>
<strong> {soccerPlayer.name} </strong>
<span> {soccerPlayer.nick} </span>
```

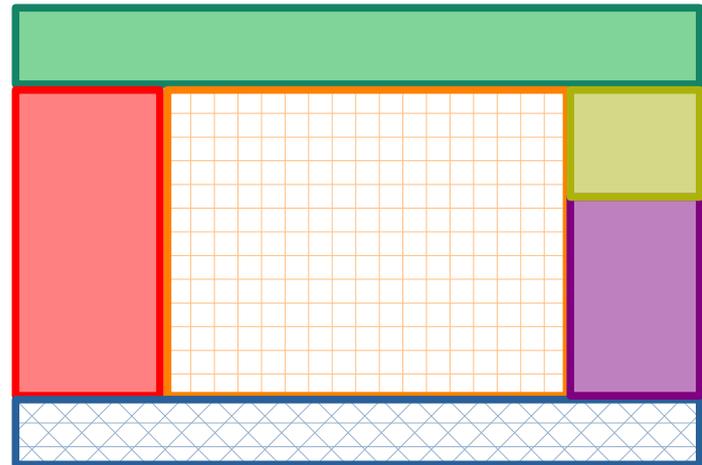
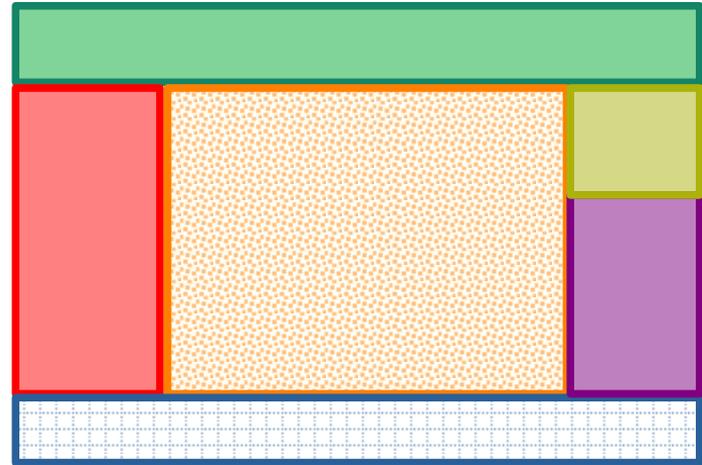
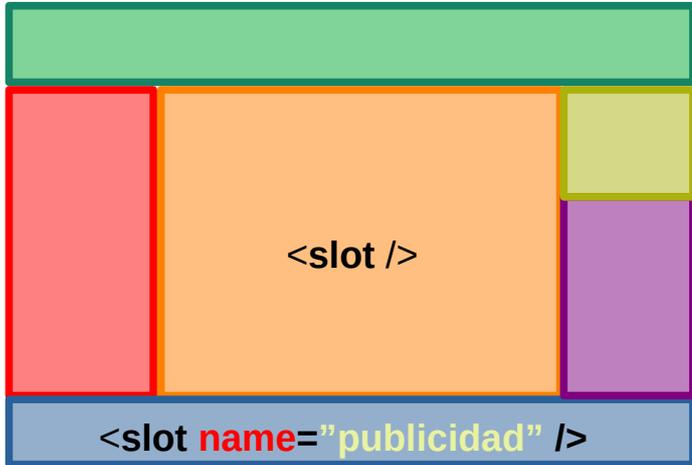
**Mi nuevo sitio Astro**

**Hola mundo**

**7 Cristiano Ronaldo CR7**

# Layouts

Definen la estructura de la **interfaz de usuario compartida** por una o más páginas



<https://docs.astro.build/en/basics/layouts>

<https://docs.astro.build/en/basics/astro-components/#named-slots>

# Layouts

Crear el archivo `src/layouts/BaseLayout.astro`

```
---  
interface Props {  
  title: string;  
}  
  
const {title="Principal"}:Props= Astro.props;  
---  
<html lang="es">  
  <head>  
    <title>{title}</title>  
  </head>  
  <body>  
    <slot name="subtitle" />  
    <slot />  
  </body>  
</html>
```

Creación de **Layout**

Crear el archivo `src/pages/example.astro`

```
---  
import BaseLayout from  
"@layouts/BaseLayout.astro";  
---  
  
<BaseLayout title="Facebook">  
  <h2 slot="subtitle">Astro</h2>  
  <p>Hola mundo</p>  
</BaseLayout>
```

Uso de **Layout** en una **Page**

# Layouts

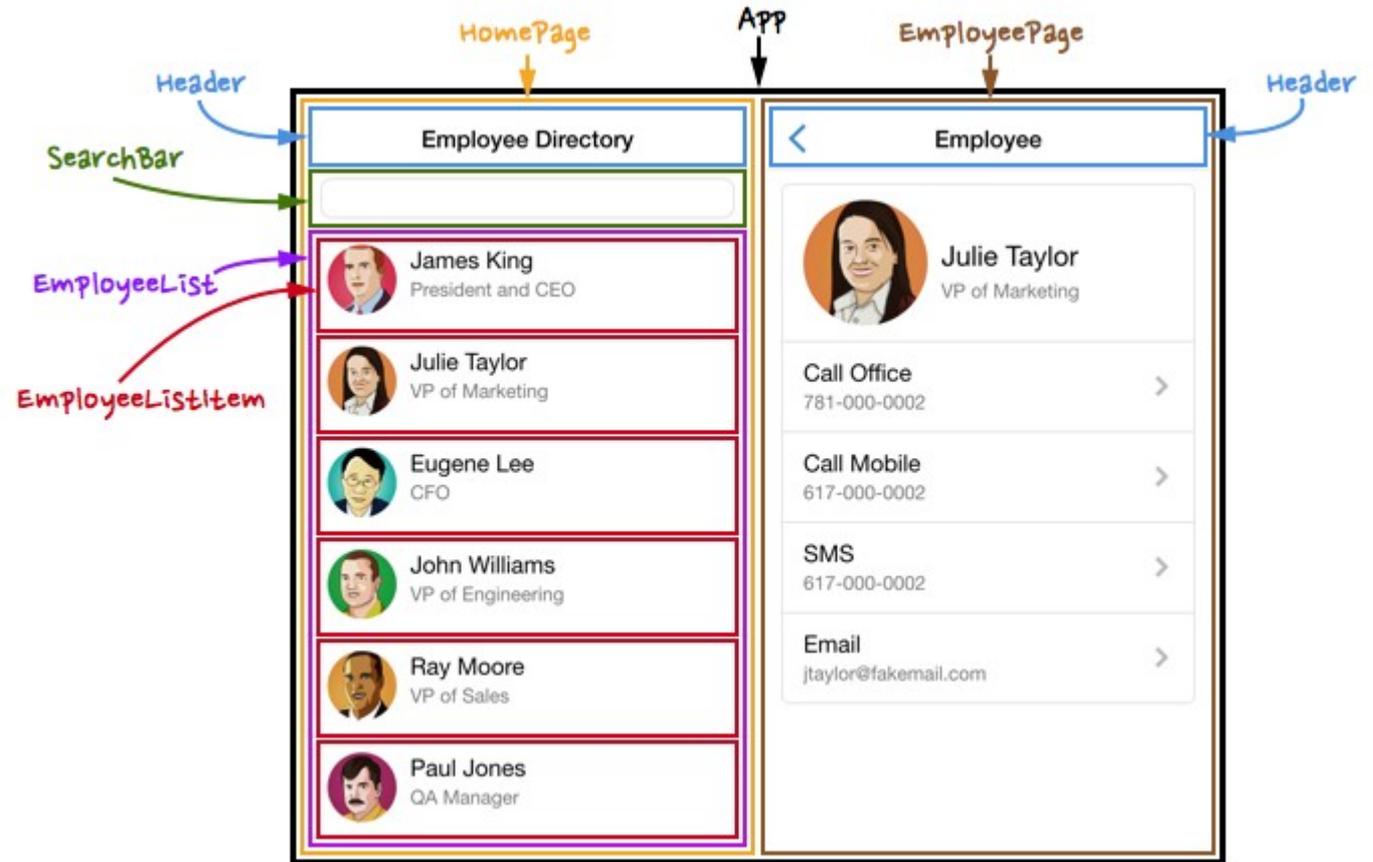
Los **Layouts** son **componentes** de Astro para proporcionar una estructura reutilizable, como una plantilla de página.

Un **<slot/>** se utiliza para especificar dónde colocar el contenido de la página.

Pueden aceptar **props** e importar y usar otros **componentes** como cualquier otro componente de Astro. Pueden incluir componentes de framework y scripts de lado del cliente.

Los componentes de **Layouts** se colocan comúnmente en la carpeta **src/layouts**

# Componentes



*Archivos reutilizables de código HTML para las paginas web.*

# Creación y uso de Componentes

Crear el archivo `src/components/Gamer.astro`

Escribir las siguientes líneas de código.

```
---  
interface Props {  
  name: string;  
  nick?: string;  
  nro: number;  
};  
const {name, nick='SN', nro}= Astro.props;  
---  
<p>{name}</p>  
<p>{nick}</p>  
<p>{nro}</p>
```

Creación de **Component**

Agregar al archivo `src/pages/index.astro`

```
---  
import Gamer from  
"@components/Gamer.astro";  
---  
<Gamer  
  name={gamer.name}  
  nick={gamer.nick}  
  nro={gamer.nro}  
>  
<Gamer name="Messi" nro=10/>
```

Uso de **Component** en una **Page**

# Renderizado de un array de objetos

```
---
import Gamer from
"@components/Gamer.astro";
const gamers = [
  {
    name: "Cristiano Ronaldo",
    nro: 25, nick: "CR7",
  },
  {
    name: "Messi", nro: 30,
    nick: "Messi",
  },
];
---
```

```
<ul>
  {
    gamers.map((gamer) => (
      <li>
        <Gamer
          name={gamer.name}
          nick={gamer.nick}
          age={gamer.nro}
        />
      </li>
    )
  )
}
</ul>
```