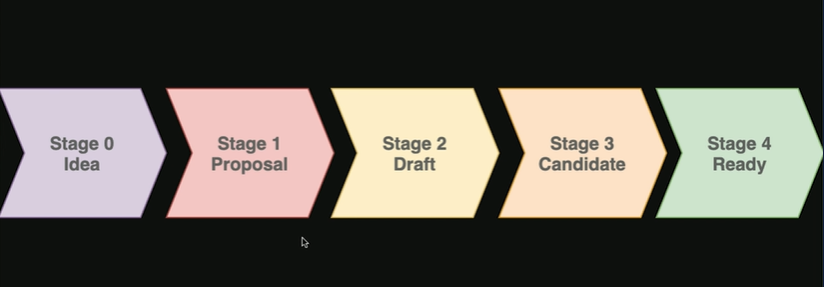
**CURSO DE ECMASCRIPT HISTORIA Y VERSIONES DE JAVASCRIPT**

**TC39** es un grupo de desarrolladores, académicos y hackers que están a cargo de revisar cada nueva propuesta o funcionalidad que cumpla con el estándar. El estándar es una serie de pasos que la nueva propuesta sigue para publicarla en la alguna versión de ECMAScript a futuro.

**Etapas de una nueva propuesta para ECMAScript**

Las etapas de una nueva propuesta para ECMAScript son:

* Idea: Una inquietud del desarrollador.
* Propuesta: Cómo y por qué la idea soluciona un problema.
* Borrador: Todo lo que implica la nueva funcionalidad detalladamente.
* Candidato: La funcionalidad es probada y desarrollada por el comité.
* Preparada: La funcionalidad está lista para ser publicada.
* Etapas que sigue una propuesta de ECMAScript
* En la página de TC39 puedes revisar qué nuevas propuestas existen y en qué etapa están.



En la [página de TC39](https://tc39.es/) puedes revisar qué nuevas propuestas existen y en qué etapa están.

**TC39:** Es un grupo de hackers, académicos y/o personas afines a la tecnología que se encargan de las actualizaciones, bajo el mando de ECMA. **ESNext** es un nombre que siempre indica la próxima versión de JavaScript y esto consta de los siguientes pasos:

* **Stage0**: Strawperson (borrador, cualquier persona puede tener una idea para implementar en el estándar)
* **Stage1**: Proposal (propuesta formal)
* **Stage2:** Draft (borrador, como va a funcionar la implementación el impacto entre otros)
* **Stage3:** Candidate (se elige el candidato, vamos a tener una propuesta que va ayudar a mejorar el lenguaje.)
* **Stage4:** Finished (va a ser desplegada en la versión normalmente en JUNIO)

**Configurando nuestras herramientas**

Para instalar los plugin que te mostré, da clic en ícono de extensiones de tu Visual Studio Code, escribe el nombre del plugin en la barra de búsqueda, selecciónalo e instala.

**Plugins importantes para este curso:**

1. **Code Runner**: Permite ejecutar piezas de código seleccionadas y muestra el output directamente en nuestro editor de código. (Más de Code Runner)
2. **Live Server**: La vamos a utilizar para ejecutar un servidor y ver los cambios efectuados en tu código. (Más de Live Server)
3. **JavaScript (ES6) code snippets**: Autocompleta y brinda sugerencias al momento de escribir código. (Más de code snippets)

**Plugins recomendados para usar en tu día a día como frontend developer**

1. **Auto Close Tag:** Ayuda a cerrar más rápido las etiquetas que vas abriendo.
2. **Error Lens:** Muestra los errores visualizando los de forma atractiva para que no pases ninguno por alto.
3. **Guides:** Te ayudan a intentar tu código y ordenarlo de forma más eficiente.
4. **Indent-rainbow:** Hace que la indotación sea más amigable con la ayuda de colores diferentes por nivel.
5. **Palenight Theme:** Úsalo si quieres visualizar los mismos colores que se muestran en mi editor de código.