**Brian Torres**

**24042974**

**CUESTIONARIO**

1. **Defina qué es un entorno virtual en Python y cuál es su importancia.**

Un entorno virtual de Python es un ambiente creado con el objetivo de aislar recursos como librerías y entorno de ejecución, del sistema principal o de otros entornos virtuales. Lo anterior significa que, en el mismo sistema, maquina o computadora, es posible tener instaladas múltiples versiones de una misma librería sin crear ningún tipo de conflicto.

Cuando se está desarrollando software con [Python](https://rukbottoland.com/blog/etiquetas/python/), es común utilizar diferentes versiones de un mismo paquete.

1. **Defina cuál es la diferencia fundamental entre los frameworks Python/Flask y Django.**

Django es un framework web Python de pila completa. Además, está desarrollado en base a las pilas incluidas. Las baterías incluidas en Django facilitan a los desarrolladores de Django realizar tareas comunes de desarrollo web como la autenticación de usuarios, el enrutamiento de URL y la migración de esquemas de bases de datos. Además, Django acelera el desarrollo de aplicaciones web personalizadas al proporcionar un motor de plantillas integrado, un sistema ORM y una herramienta de arranque. Por otro lado, Flask es un marco web simple, ligero y minimalista. Carece de algunas de las características integradas proporcionadas por Django. Pero ayuda a los desarrolladores a mantener el núcleo de una aplicación web simple y extensible.

1. **Defina qué es Django.**

es un [framework](https://es.wikipedia.org/wiki/Framework" \o "Framework) de desarrollo web de [código abierto](https://es.wikipedia.org/wiki/Open_Source), escrito en [Python](https://es.wikipedia.org/wiki/Python), que respeta el patrón de diseño conocido como [Modelo–vista–template](https://docs.djangoproject.com/en/2.0/). La meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio [No te repitas](https://es.wikipedia.org/wiki/No_te_repitas) (DRY, del inglés Don't Repeat Yourself). Python es usado en todas las partes del framework, incluso en configuraciones, archivos, y en los modelos de datos.

1. **Defina ASP.NET y ASP.NET MVC.**

Es un [entorno para aplicaciones web](https://es.wikipedia.org/w/index.php?title=Entorno_para_aplicaciones_web&action=edit&redlink=1) desarrollado y comercializado por [Microsoft](https://es.wikipedia.org/wiki/Microsoft). Es usado por programadores y diseñadores para construir [sitios web](https://es.wikipedia.org/wiki/Sitio_web) dinámicos, [aplicaciones web](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web) y [servicios web](https://es.wikipedia.org/wiki/Servicio_Web) [XML](https://es.wikipedia.org/wiki/XML). Apareció en enero de 2002 con la versión 1.0 del [.NET Framework](https://es.wikipedia.org/wiki/.NET_Framework), y es la tecnología sucesora de la tecnología [Active Server Pages](https://es.wikipedia.org/wiki/Active_Server_Pages) (ASP). ASP.NET está construido sobre el [Common Language Runtime](https://es.wikipedia.org/wiki/Common_Language_Runtime" \o "Common Language Runtime), permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el [.NET Framework](https://es.wikipedia.org/wiki/.NET_Framework).

**ASP.NET MVC** se concibió como alternativa a Web Forms y proporciona un modelo de programación basado en el popular patrón de arquitectura MVC. Entre sus principales características destacan su completa integración con pruebas unitarias y su separación más clara entre la lógica de presentación, la lógica de negocio y la lógica de acceso a datos.

1. **Describa las diferencias entre ASP.NET y Django.**

Generalmente un framework puede ser limitado en funcionalidad, pero de alto desempeño o especialización. Otra posibilidad es la disponibilidad de muchas bibliotecas (probablemente de terceros) que pueden no funcionar o requerir algunas modificaciones.

Generalmente .NET, al ser un framework desarrollado por Microsoft, es el primer caso y Django y Ruby son el segundo. Qué opción es mejor depende de un proyecto, los plazos y la competencia en las áreas relacionadas con la web