TECNICAS DE DISEÑO

ALUMNA: KATERIN MERINO QUISPE

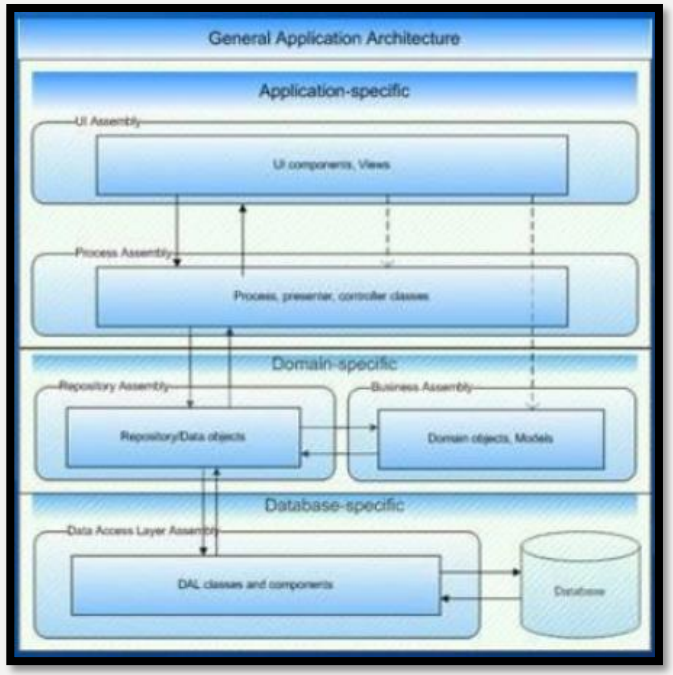
DDD(DOMAIN-DRIVEN DESIGN)

- Es una forma de diseñar el software centrándonos en lo que el cliente nos pide. El software que hacemos tiene como objetivo resolver un problema de nuestro cliente.
- DDD es un estilo arquitectural. Se parece bastantea un estilo en N-Layer en tanto que los dos estilos separan las responsabilidades de la aplicación, pero la forma de hacerlo es ligeramente distinta.

DIAGRAMA

CAPAS Y NIVELES(Layers vs. Tiers)

- Es la forma de organizar el código lógicamente y no físicamente.



- Es la separación física en diferentes proyectos.

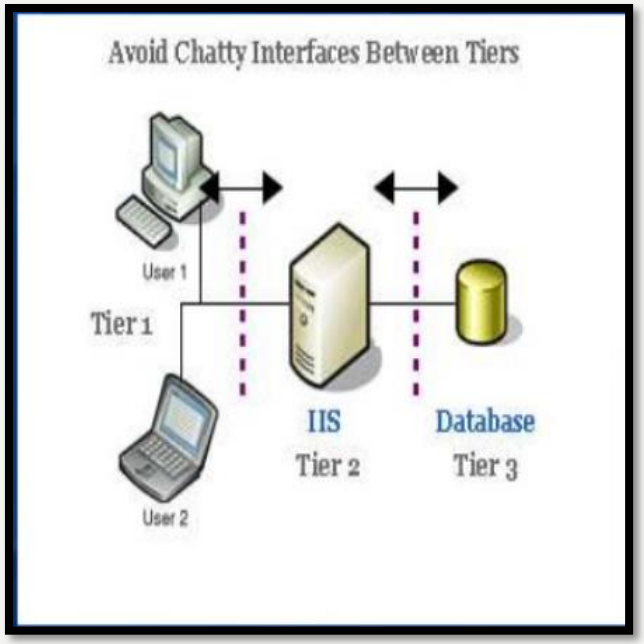
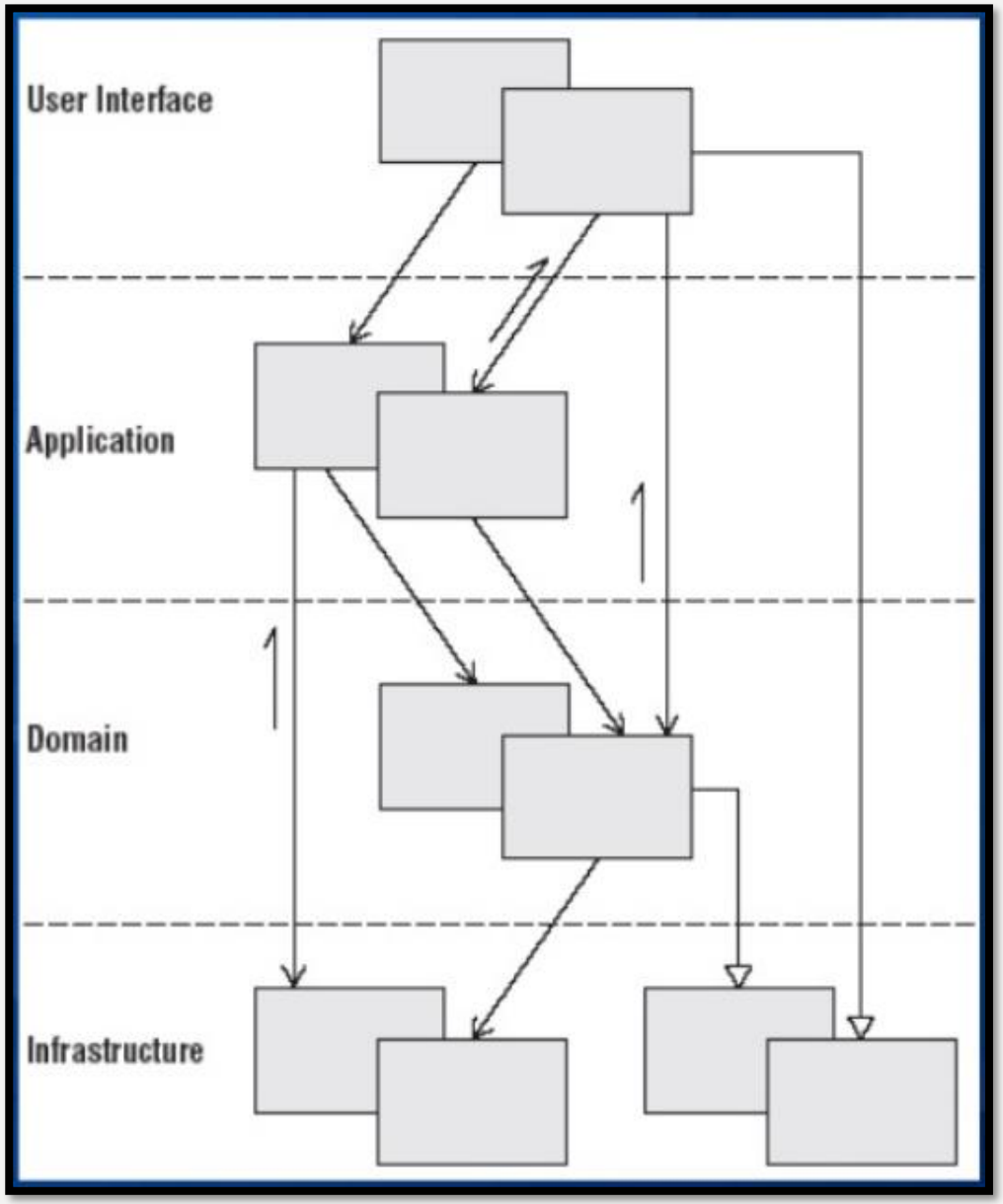


DIAGRAMA DE EVANS

- La mas fácil de entender, esta capa es la responsable de mostrar información al usuario, y aceptar nuevos datos. Puede ser implementada para web.
- Como: WEB: ASP.NET, ASP.NET MVC; Winforms; WPF; Silverlight.

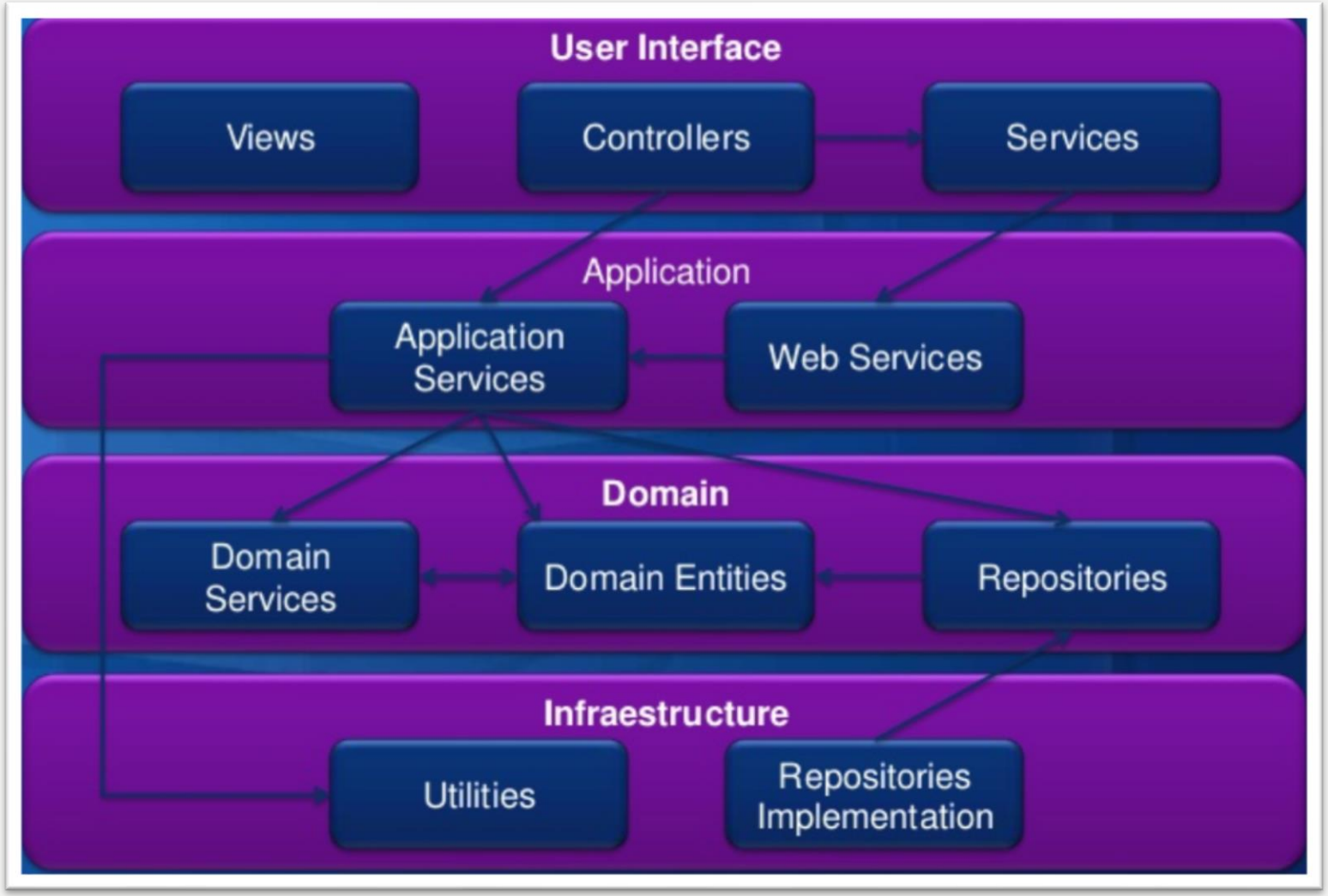


- Define los trabajos que el software debe realizar y dirige los objetos del dominio para que trabajen en cada problema.
- Es delgada, no contiene reglas de negocio o conocimiento.

- En esta capa reside el corazon del software, según Evans. Las reglas y lógica de negocio residen en esta capa. Como: Entities, Value Objects, Repositories, Services y Factories.

- Esta capa es la responsable de implementar el mecanismo de persistencia del modelo de dominio y de proporcionar la implementación de todas las operaciones de comunicación.

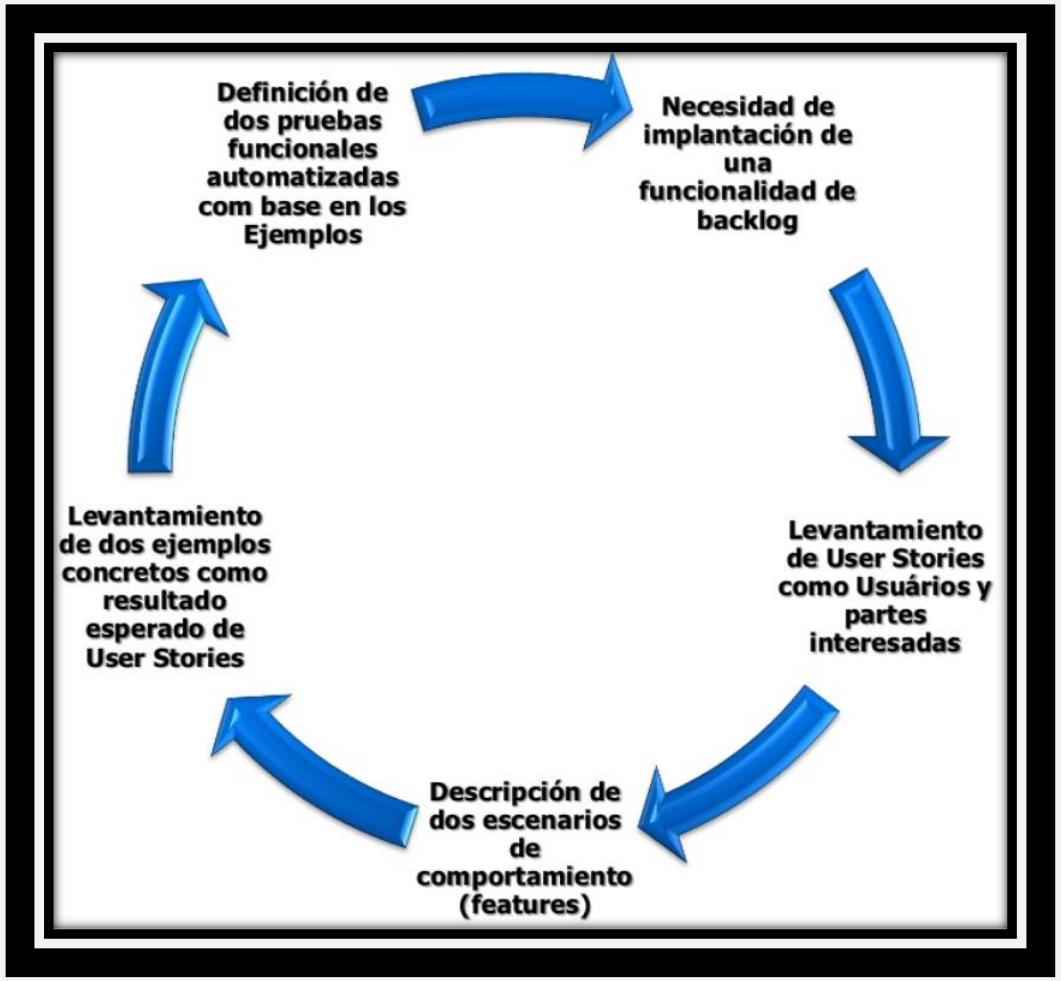
DIAGRAMA SUGERIDO



BDD(BEHAVIOUR DRIVEN DESIGN)

- Es un conjunto de prácticas ágiles para agilizar el desarrollo de software a través de la integración de las User Stories definidas para el software con la automatización de las pruebas funcionales del software desarrollado.
- El BDD es una evolución hecha a partir de la implantación de técnicas TDD (Test Driven Development).
- El BDD no es una metodología de desarrollo de software, pero incorpora y mejora las ideas de muchas de esas metodologías.

- FUNCIONAMIENTO:

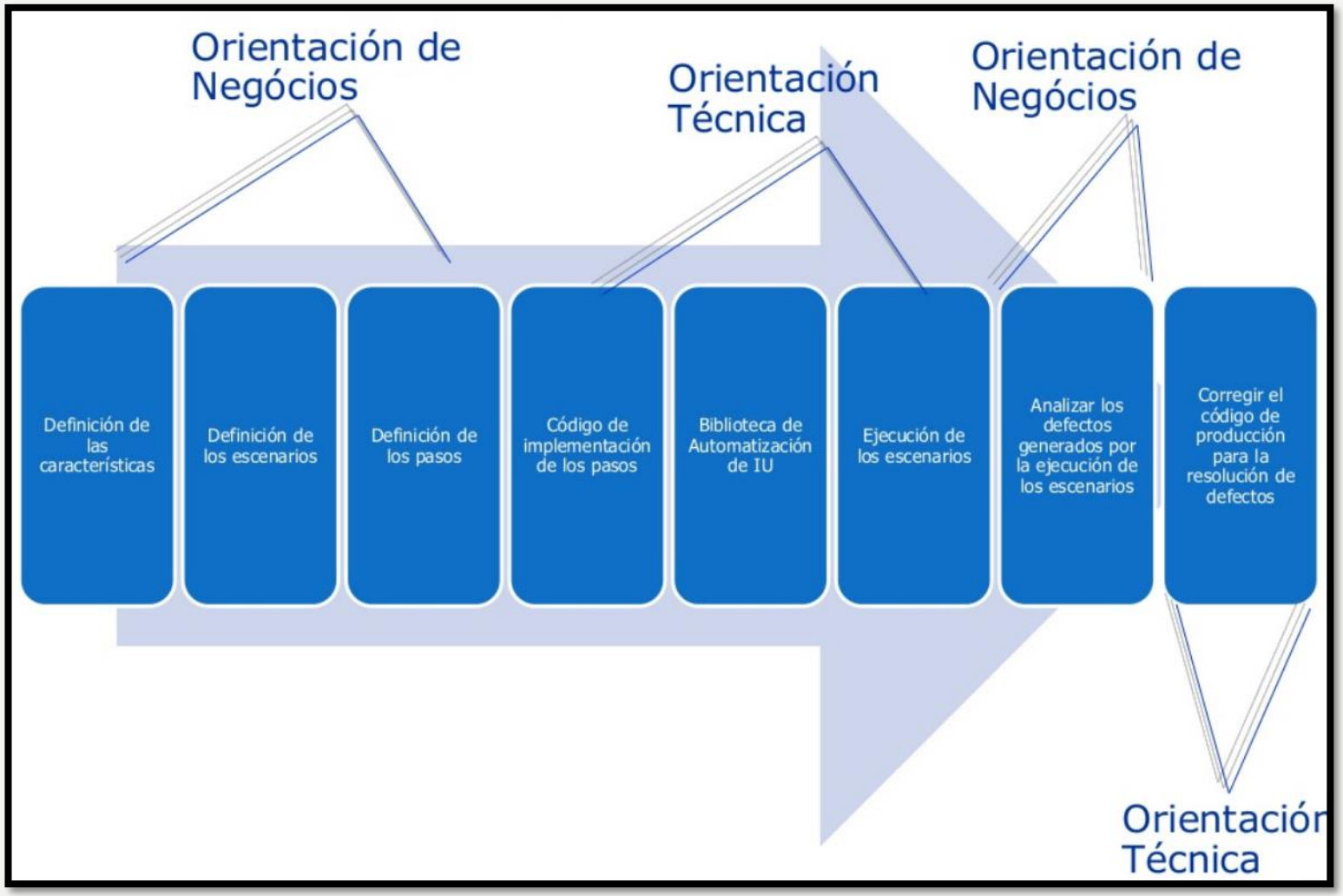


DIFERENCIAS FUNDAMENTALES DEL BDD X TDD

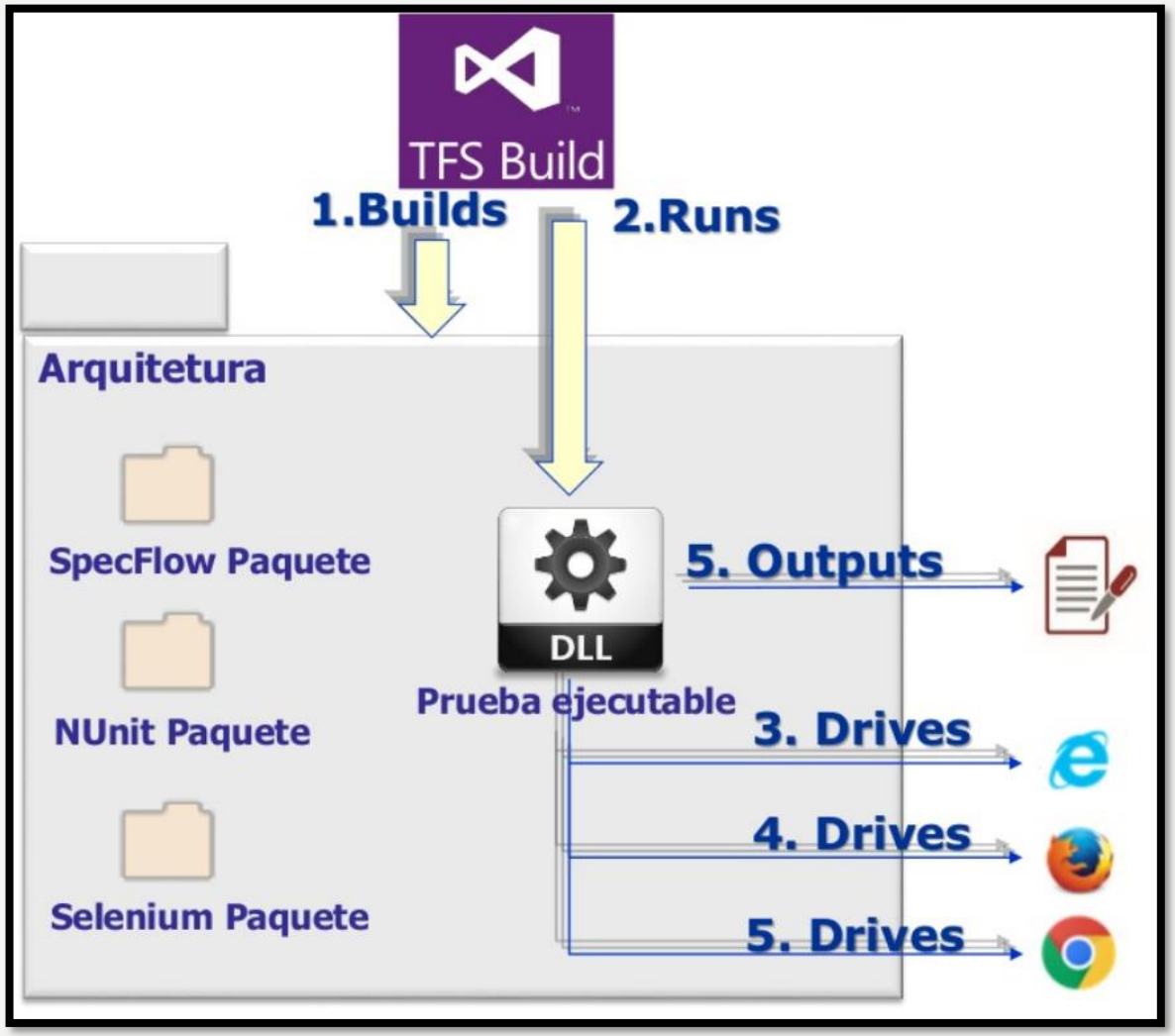
- **TDD-** Ejercita el código vinculado a cada método del objeto.

- **BDD** -valida si el resultado presentado por la UI está de acuerdo con el comportamiento definido en los ejemplos en el User Story.

DIAGRAMA DE SPECFLOW

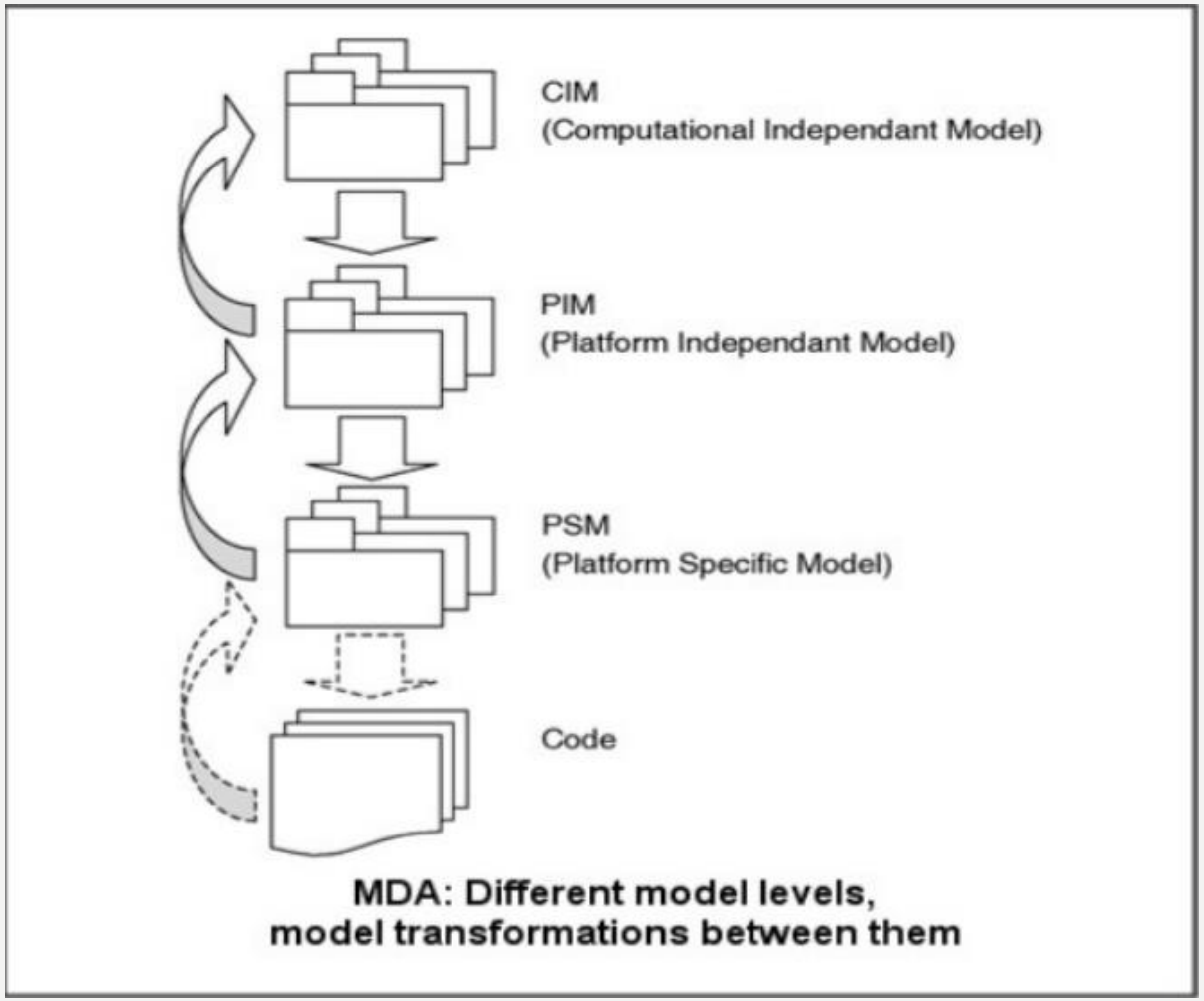


ARQUITECTURA .NET DE EJCUCION CON CONTINUOS DELIVERY



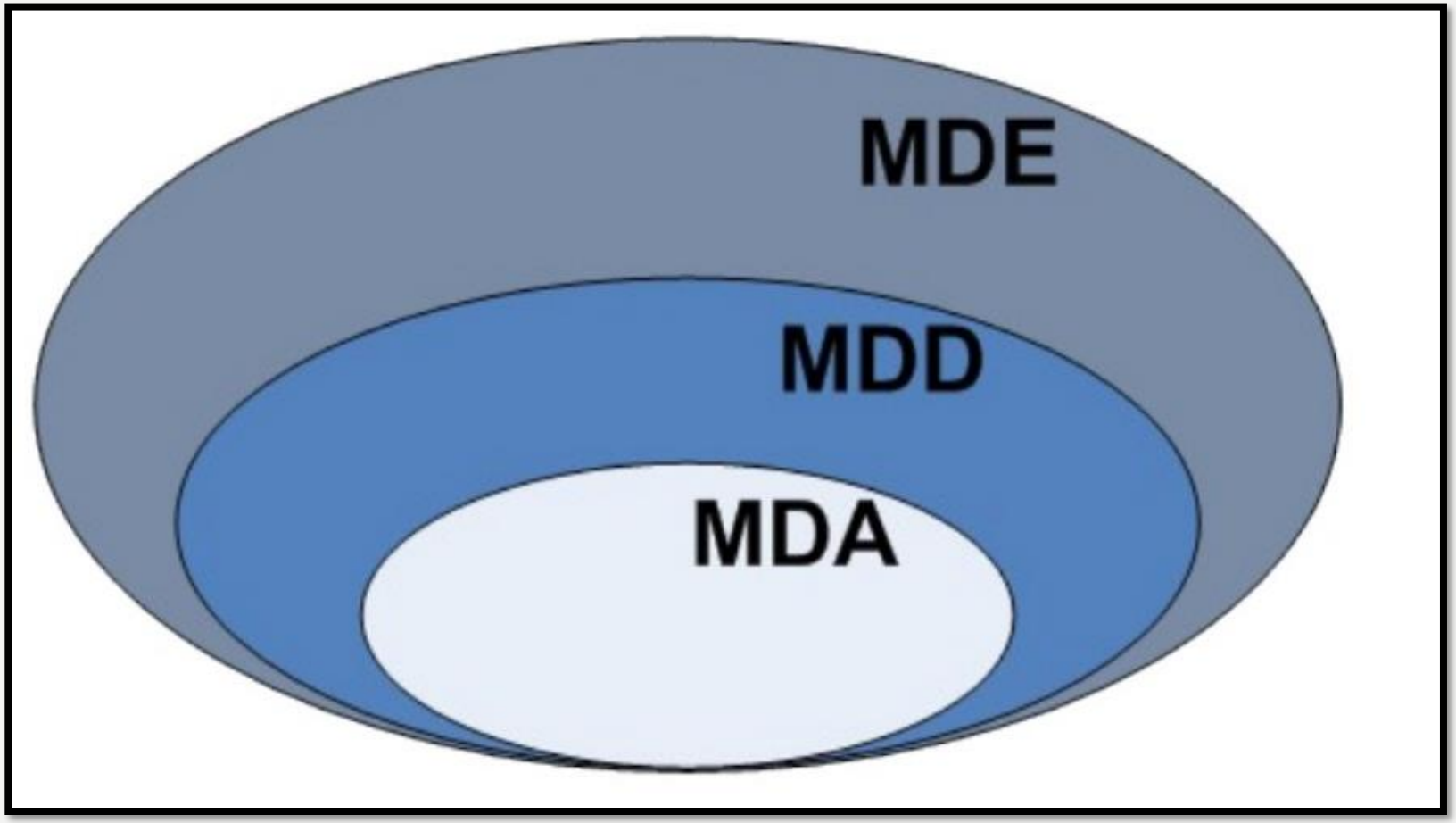
MDD(DESARROLLO DE SOFTWARE DIRIGIDO POR MODELOS QUE FUNCIONA)

- Puede verse como un conjunto de transformaciones modelo a modelo (M2M) mas una transformación final modelo a texto(M2T)

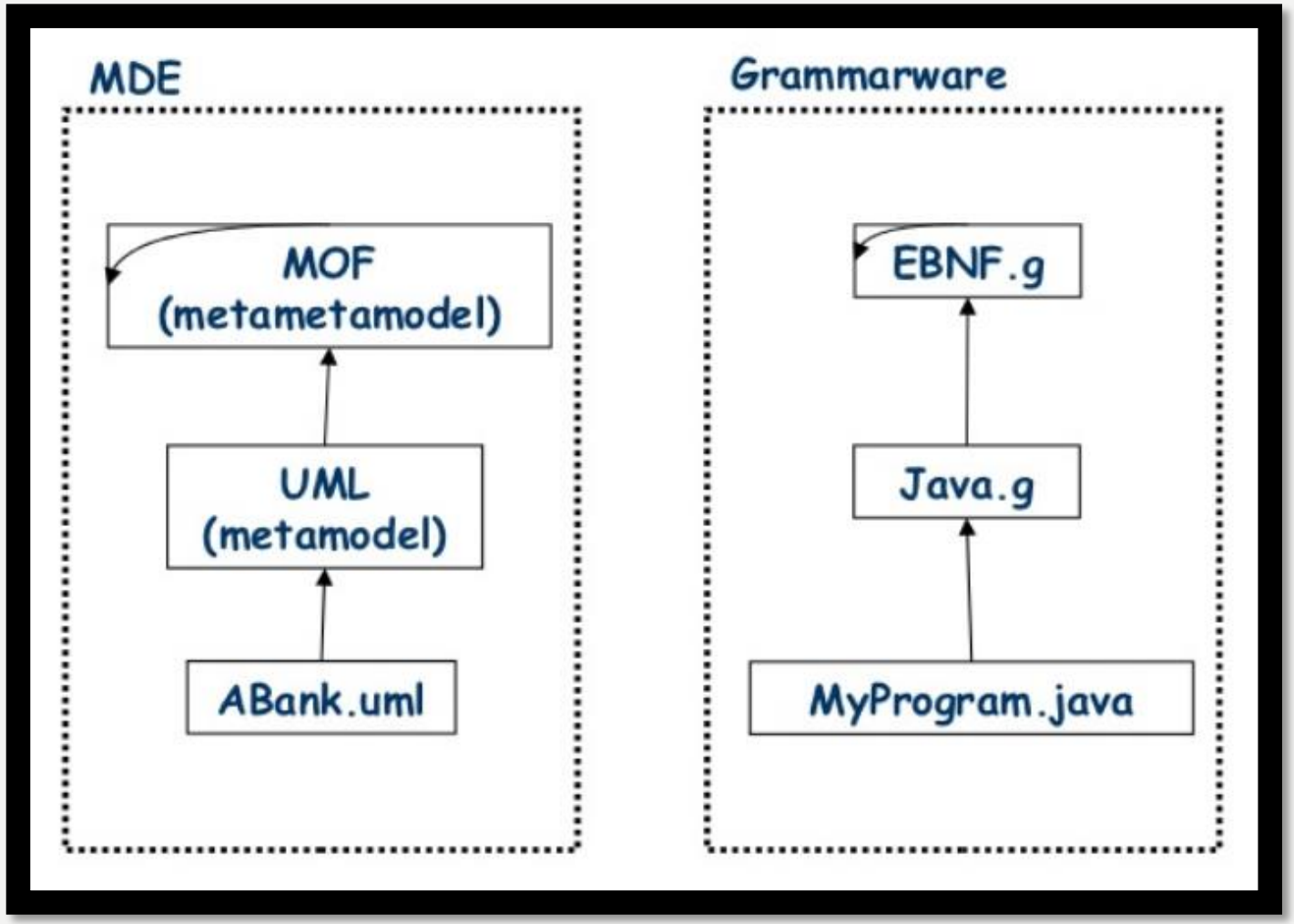


MDA: Different model levels, model transformations between them

RELATIONSHIT BETWEEN MDA/MDD/MDE



ELEMENTOS CLAVE EN MDD: MODELOS Y TRANSFORMACIONES



CONCLUSIONES

- El diseño detallado nos da una visión mas amplia sobre el lenguaje que se utilizará para el desarrollo del sistema, gracias al diseño detallado podemos prever errores.
- Así como la definición de algoritmos y sentencias a utilizar.

GRACIAS