Instituto Tecnológico Superior de Purísima del Rincón

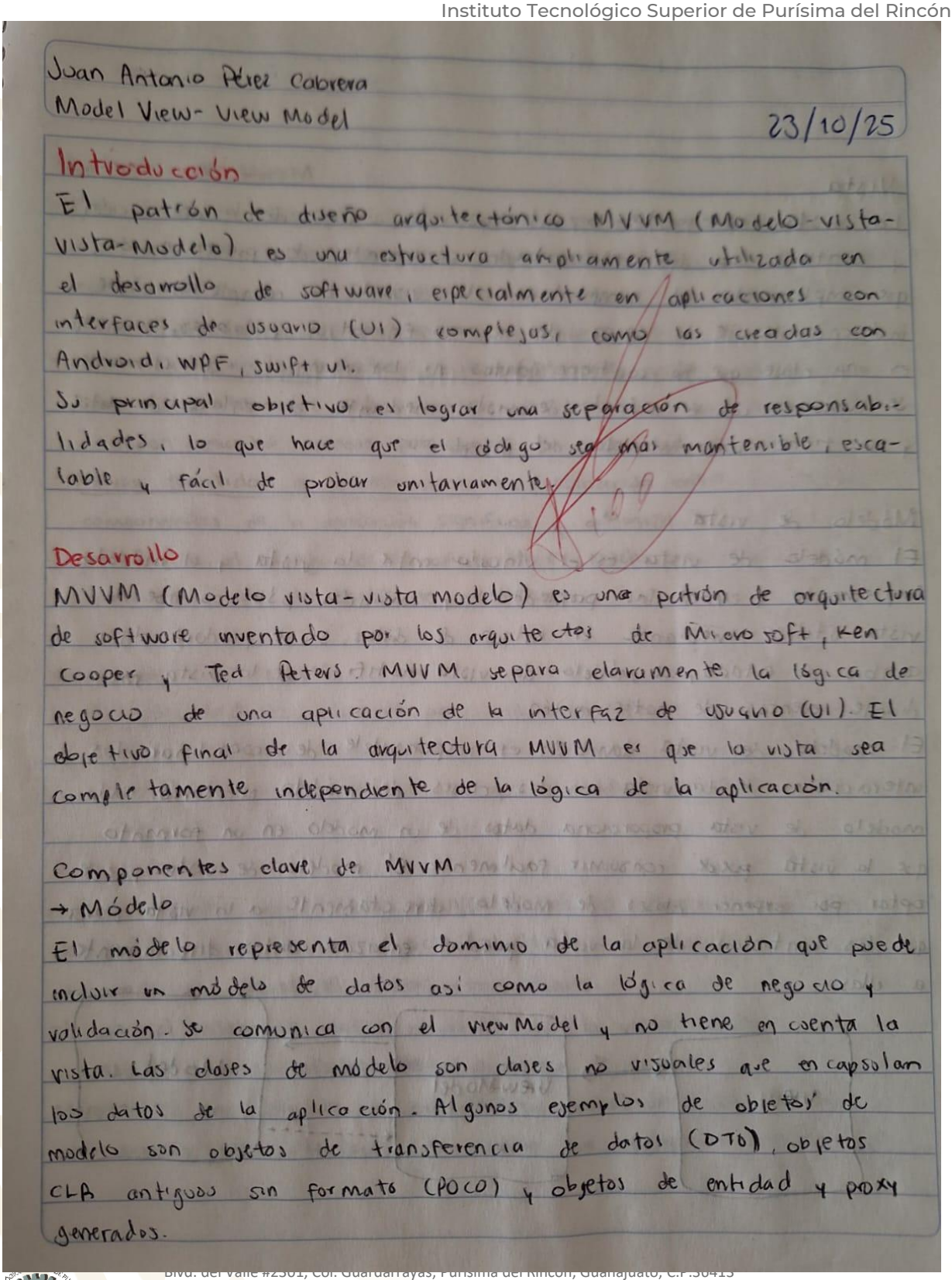
T1_Model View - View Model

MATERIA:
Desarrollo de aplicaciones para dispositivos móviles

ALUMNOS:
Juan Antonio Pérez Cabrera

DOCENTE:
Jair Emmanuel Ramírez Flores

FECHA: 23 de octubre de 2025 **CIUDAD:** Purísima del Rincón, Gto.



Juan Antonio Pérez Cabrera
Model View- View Model
23/10/25

Introducción

El patrón de diseño arquitectónico MVVM (Modelo-vista-vista-Modelo) es una estructura ampliamente utilizada en el desarrollo de software, especialmente en aplicaciones con interfaces de usuario (UI) complejos, como las creadas con Android, WPF, swift ui.

Su principal objetivo es lograr una separación de responsabilidades, lo que hace que el código sea más mantenible, escalable y fácil de probar unitariamente.

Desarrollo

MVVM (Modelo vista-vista modelo) es una patrón de arquitectura de software inventado por los arquitectos de Microsoft, Ken Cooper y Ted Peters. MVVM separa elaramente la lógica de negocio de una aplicación de la interfaz de usuario (UI). El objetivo final de la arquitectura MVVM es que la vista sea completamente independiente de la lógica de la aplicación.

Componentes clave de MVVM

→ Módelo

El módelo representa el dominio de la aplicación que puede incluir un módelo de datos así como la lógica de negocio y validación. Se comunica con el viewModel y no tiene en cuenta la vista. Las clases de módelo son clases no visuales que encapsulan los datos de la aplicación. Algunos ejemplos de objetos de modelo son objetos de transferencia de datos (DTO), objetos CLR antiguos sin formato (POCO) y objetos de entidad y proxy generados.

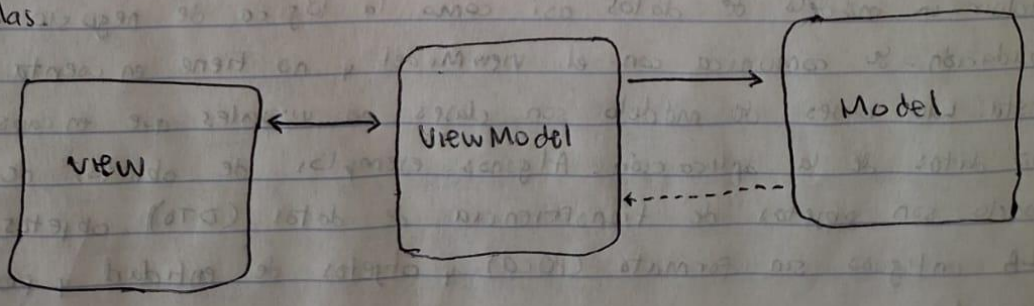
Vista

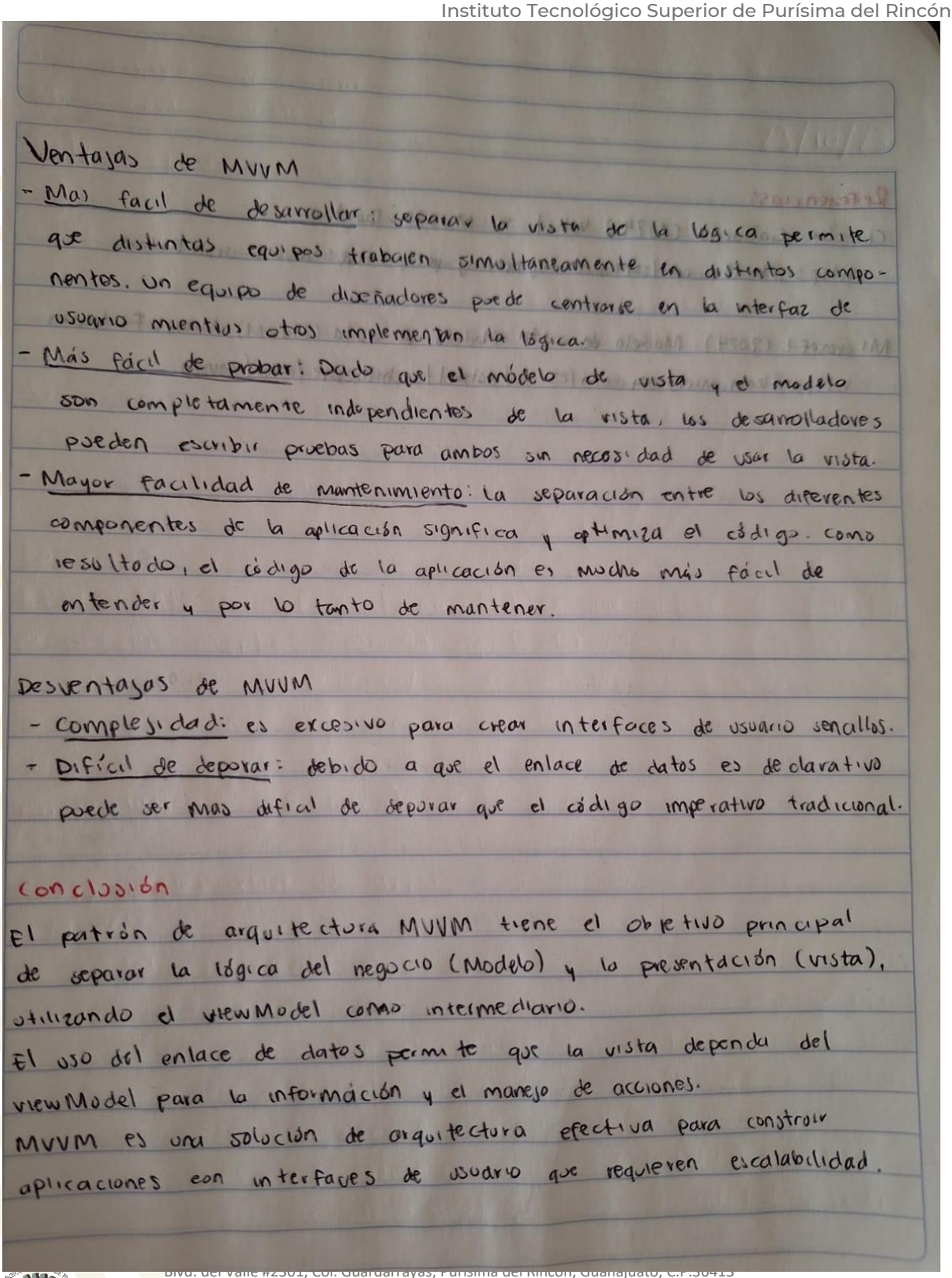
La vista representa la interfaz de usuario de la aplicación y contiene una lógica limitada, puramente de presentación, que implementa el comportamiento visual. Es completamente independiente de la lógica de negocio. En otras palabras, la vista es una clase que no contiene datos ni los manipula directamente. Se comunica con el módelo de vista, mediante el enlace de datos y no tiene conocimiento del módelo.

Módelo de vista

El módelo de vista es el vínculo entre la vista y el módelo. Implementa y expone las propiedades y comandos públicos que la vista utiliza mediante el enlace de datos. Si se produce algún cambio de estado, el módelo de vista notifica a la vista mediante eventos de notificación.

El modelo de vista también es responsable de coordinar las interacciones de la vista con las clases de módelo necesario. Cada modelo de vista proporciona datos de un modelo en un formato que la vista puede consumir facilmente. El modelo de vista puede optar por exponer clases de modelo directamente a la vista para que los controlen de la vista puedan enlazar datos directamente a ellas.





Ventajas de MVVM

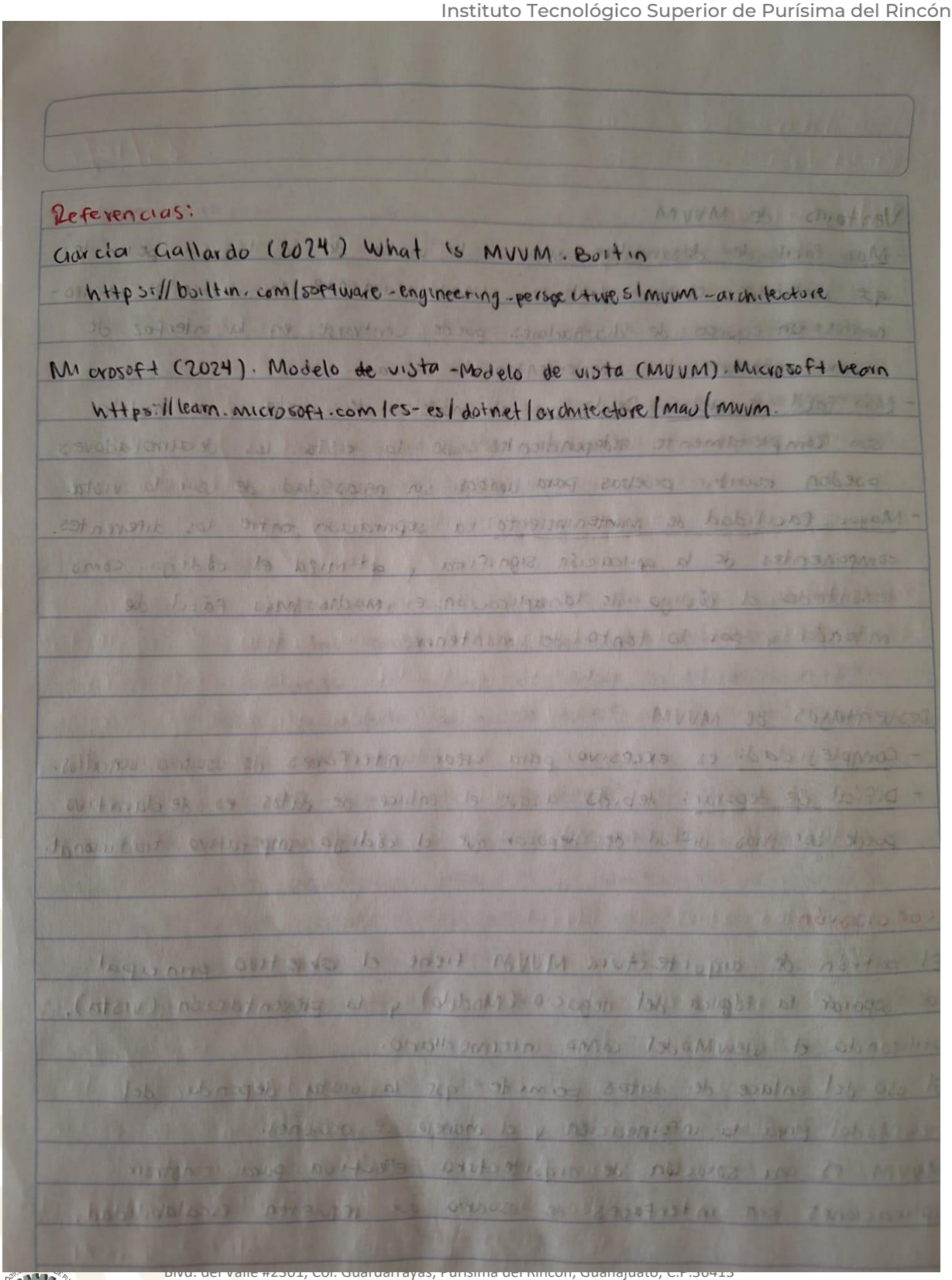
- Mas facil de desarrollar: separar la vista de la lógica permite que distintas equipos trabajen simultaneamente en distintos componentes. Un equipo de diseñadores puede centrarse en la interfaz de usuario mientras otros implementan la lógica.
- Más fácil de probar: Dado que el módelo de vista y el modelo son completamente independientes de la vista, los desarrolladores pueden escribir pruebas para ambos sin necesidad de usar la vista.
- Mayor facilidad de mantenimiento: La separación entre los diferentes componentes de la aplicación significa y optimiza el código. como resultado, el código de la aplicación es mucho más fácil de entender y por lo tanto de mantener.

Desventajas de MVVM

- Complejidad: es excesivo para crear interfaces de usuario sencillos.
- Difícil de deporar: debido a que el enlace de datos es declarativo puede ser mas dificil de deporar que el código imperativo tradicional.

conclusión

El patrón de arquitectura MVVM tiene el objetivo principal de separar la lógica del negocio (Modelo) y la presentación (vista), utilizando el viewModel como intermediario.
El uso del enlace de datos permite que la vista dependa del viewModel para la información y el manejo de acciones.
MVVM es una solución de arquitectura efectiva para construir aplicaciones con interfaces de usuario que requieren escalabilidad.



Referencias:

García Gallardo (2024) What is MVVM. Boltin
https://boltin.com/software-engineering-perspectives/mvvm-architecture

Microsoft (2024). Modelo de vista - Modelo de vista (MVVM). Microsoft Learn
https://learn.microsoft.com/es-es/dotnet/architecture/maui/mvvm.