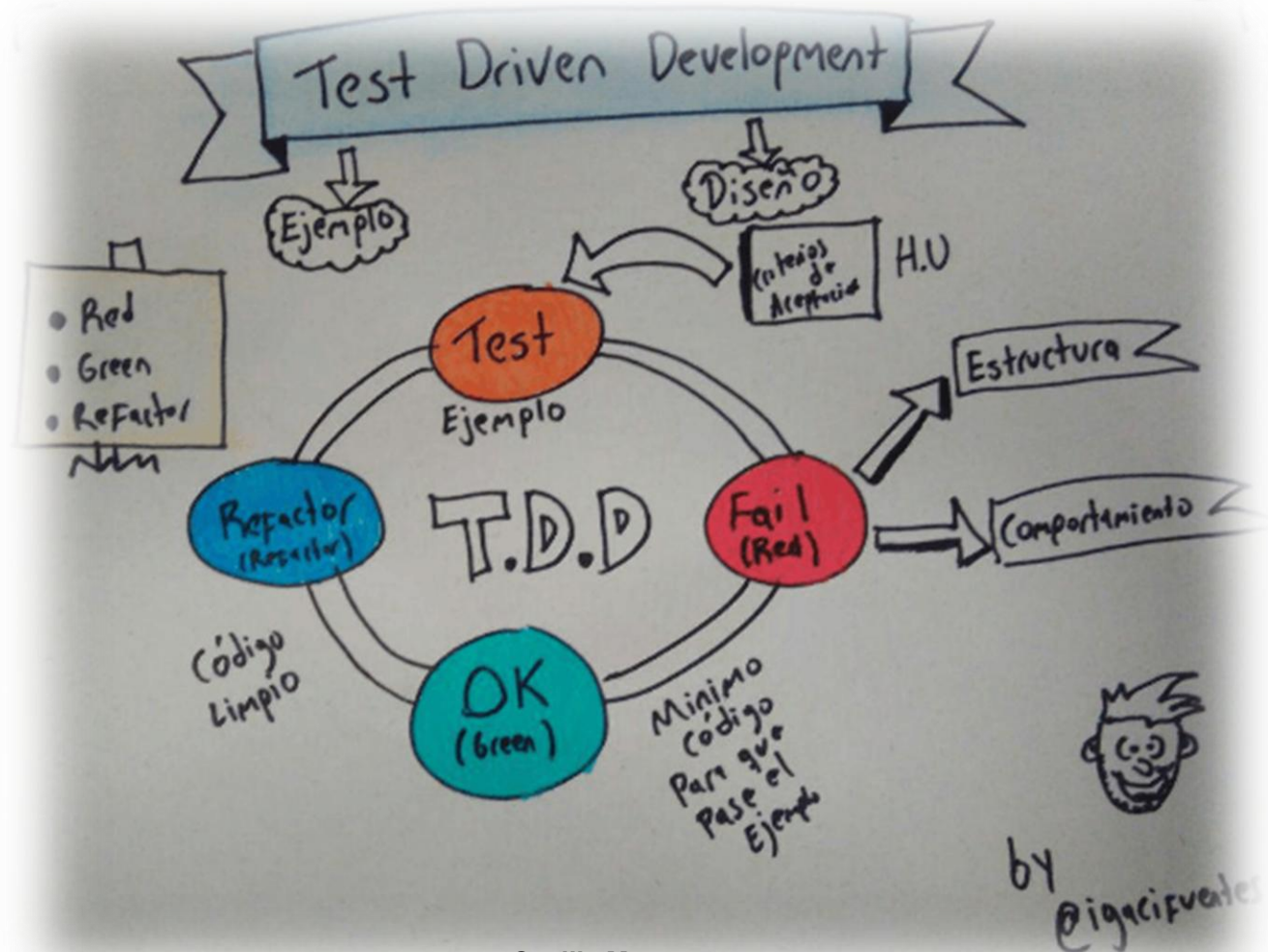
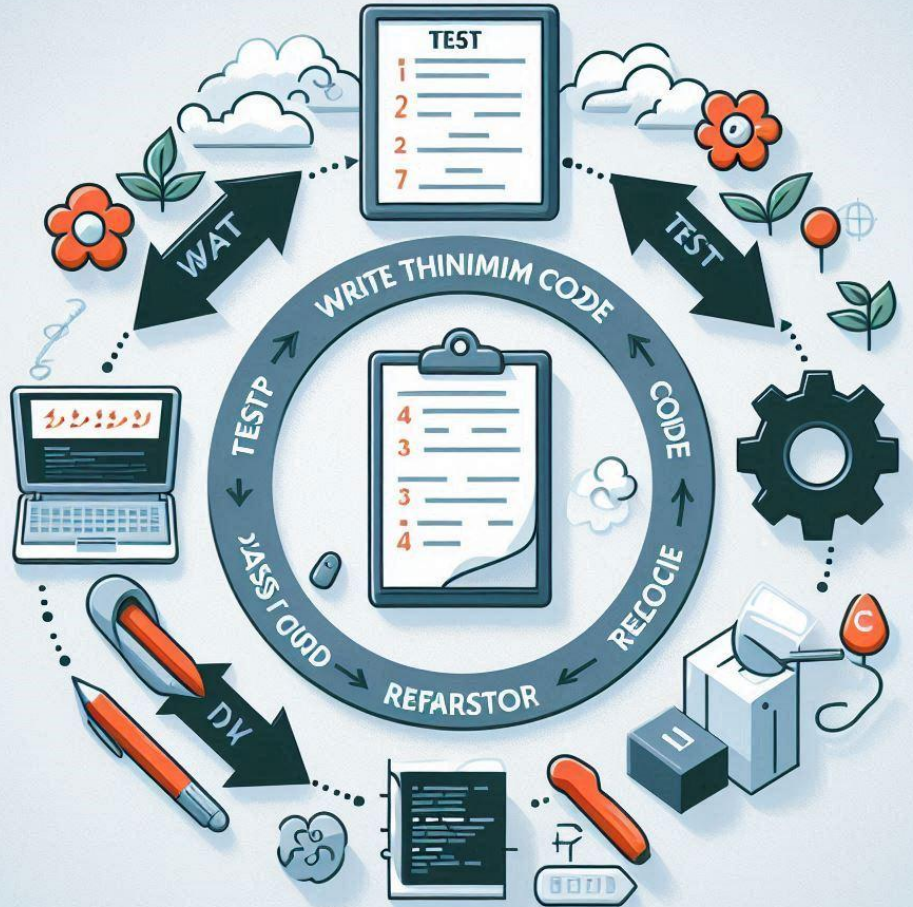


Test Driven Development (TDD)



Práctico de TDD Taxi Mobile





// PRECONDICIONES

Establecen el **estado inicial** necesario para ejecutar correctamente los pasos de prueba.

def: Es una palabra clave que define una función. En este caso, la función llamada

Ejemplo-Taxi Mobile:

Taxis Libres (identificador verde) con chapa válida para un Barrio existente

```
def test_mostrar_mapa_taxis_libres_en_barrio ():  
    // PRECONDICIONES  
    usuario = {'nombre': 'Franco', 'apellido':  
        'Colapinto', 'permisos': 'Administrador de la  
        Central'}  
    usuario = iniciar_sesion('admin','clave_admin')  
    configurar_software('Ciudad de Córdoba')  
    taxis = [{'chapa': 'ABC123', 'estado': 'Libre',  
        'barrio': 'Nueva Córdoba', 'ubicacion': (10, 20)}]
```



// PASOS DEL CASO DE PRUEBA

Estos pasos reflejan interacciones del usuario con el sistema y realizan el llamado a la US

mostrar_mapa(...): Llama a la funcionalidad principal de la **Historia de Usuario (US)** y envía los parametros, busca mostrar taxis en el mapa filtrados por estado y barrio.

Ejemplo-Taxi Mobile:

Taxis Libres (identificador verde) con chapa válida para un Barrio existente

// PASOS DEL CASO DE PRUEBA

```
seleccionar_opcion_menu('Ver mapa de taxis')  
// Llamado a la US  
mostrar_mapa (opcion = 'Mostrar taxis por  
estado y por barrio', barrio = 'Nueva  
Córdoba', chapa = [], estado = 'Libre')
```



// RESULTADOS

Define qué debe devolver el sistema y valida que el comportamiento es correcto.

assert: Es una palabra clave que se usa para verificar condiciones en las pruebas. Si la condición dentro del **assert** es falsa, la prueba fallará.

assert all: Implica que se deben cumplir todas las condiciones para que cumpla la prueba

Ejemplo-Taxi Mobile:

Taxis Libres (identificador verde) con chapa válida para un Barrio existente

// RESULTADOS

```
mapa_mostrado = mostrar_mapa()
assert mapa_mostrado['ciudad'] == 'Córdoba',
"El mapa mostrado no corresponde a la ciudad
de Córdoba"
assert all(taxi['estado'] == 'Libre' for taxi
in mapa_mostrado['taxis']), "No todos los
taxis están en estado 'Libre'"
assert all(taxi['icono'] == 'verde' for taxi
in mapa_mostrado['taxis']), "No todos los
taxis tienen el ícono verde"
```



Mensaje final

Si todas las verificaciones anteriores con `assert` son verdaderas, se imprimirá este mensaje, indicando que la prueba ha sido exitosa.

Ejemplo-Taxi Mobile:

Taxis Libres (identificador verde) con chapa válida para un Barrio existente

```
//Este print sale si todos los casos son verdaderos
```

```
print("La prueba para mostrar taxis libres en el barrio 'Nueva Córdoba' ha pasado.")
```



PSEUDOCÓDIGO DE LA US

El método `mostrar_mapa` recibe cuatro parámetros: `opcion`, `barrio`, `chapa`, `estado` y `ubicación`, y luego toma decisiones basadas en las diferentes opciones para mostrar los taxis.

Ver mapa de taxis

Como Administrador de la Central quiero ver la ubicación de todos los taxis de la central y si tienen viajes en curso para saber la disponibilidad actual.

1. Probar visualizar un conjunto de taxis libres con una chapa válida para un barrio en particular(pasa)
2. Probar visualizar un conjunto de taxis libres, solicitados, fuera de servicio y ocupados, de un barrio en particular(pasa)
3. Probar visualizar un taxi conectado al sistema por número de chapa (pasa)
4. Probar visualizar un conjunto de taxis ocupados (pasa)
5. Probar visualizar los datos de un taxi solicitado y el pasajero que lo solicitó (pasa)
6. Probar visualizar un taxi en un Barrio inexistente (falla)
7. Probar visualizar el conjunto de taxis fuera de servicio (pasa)
8. Probar visualizar los datos de un taxi ocupado y el pasajero que lo solicitó (pasa)

Ejemplo-Taxi Mobile:

Taxis Libres (identificador verde) con chapa válida para un Barrio existente

```
def mostrar_mapa(opcion, barrio, chapa, estado, ubicación):
```

```
    // Caso de prueba taxis desconectados
```

```
    if (opcion == 'Mostrar todos los taxis')
```

```
        taxis = buscarTaxiEnBarrio(barrio)
```

```
        (ubicación == None)
```

```
        taxis = buscarTaxiDesconectado (ubicacion)
```

```
    // Caso de prueba mostrar taxis en distintos estados en un barrio
```

```
    if (opcion == 'Mostrar taxis por estado y por barrio')
```

```
        taxis = buscarTaxiPorEstadoYPorBarrio (barrio, estado)
```

```
    // Caso de prueba mostrar taxis por chapa
```

```
    if (opcion == 'Mostrar taxis por chapa')
```

```
        taxis = buscarTaxiPorChapa (chapa)
```

```
    // Caso de prueba mostrar taxis por barrio y datos de viaje
```

```
    /if (opcion == 'Mostrar taxis por barrio')
```

```
        taxis = buscarTaxiEnBarrio (barrio)
```

```
    /if (taxis[i].getEstado() == 'Ocupado' && opcion ==  
        'Mostrar datos viaje')
```

```
        taxis[i].getChapa()
```

```
        taxis[i].getViajeActual().getPasajero().getNombre()
```

```
        taxis[i].getViajeActual().getPasajero().getTelefono()
```

```
        taxis[i].getViajeActual().getFechaYHora()
```