

***Universidad Tecnológica Nacional***

***Ingeniería en sistemas de información***

**Cátedra:** Diseño de sistemas (DSI).

**Docentes:** Meles, Silvia Judith.  
 Bene, Florencia.

**Curso:** 3k4

**Grupo nro.:** 7

**Entrega nro.:** “3: Arquitectura”.

**Integrantes:**

**Fecha Entrega:**

**Índice**

Contenido

[**CONSIGNA** 4](#_Toc18070531)

[**ENUNCIADO** 6](#_Toc18070532)

[**E.R.S.** 8](#_Toc18070533)

[**ANÁLISIS** 10](#_Toc18070534)

[**Realización de caso de uso de análisis. Caso de uso 17 “Registrar pedido”** 11](#_Toc18070535)

[**Realización de caso de uso de análisis. Caso de uso 39 “Generar remito”** 12](#_Toc18070536)

[**Realización de caso de uso de análisis. Caso de uso 85 “Generar informe de efectividad”** 13](#_Toc18070537)

[**Máquina de estados. Clase “Remito”** 14](#_Toc18070538)

[**ENTREGAS CON CORRECCIONES APLICADAS** 15](#_Toc18070539)

[**MODELO DE DOMINIO** 16](#_Toc18070540)

[**ARQUITECTURA** 17](#_Toc18070541)

[**Listado y descripción de Requerimientos No Funcionales** 18](#_Toc18070542)

[**Patrones arquitectónicos** 19](#_Toc18070543)

[N TIER 20](#_Toc18070544)

[MESSAGING 20](#_Toc18070545)

[PROCESS COORDINATOR 21](#_Toc18070546)

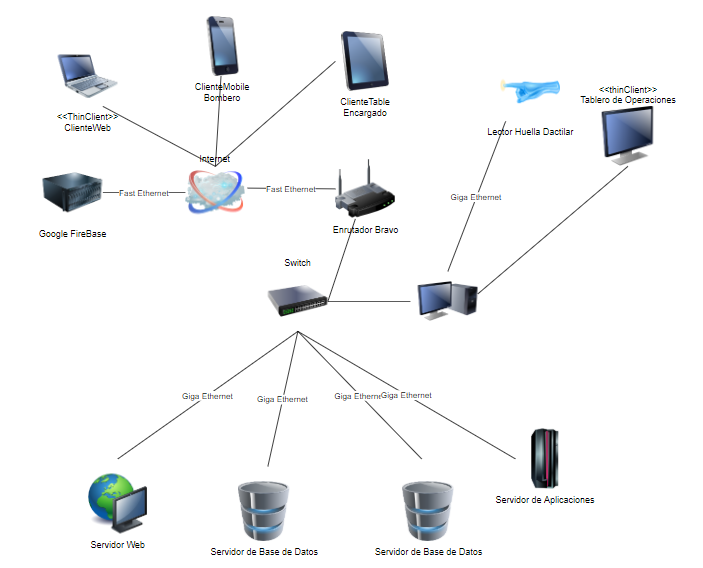
[**Vista de la funcionalidad y justificación de casos de uso** 22](#_Toc18070547)

[Vista de la funcionalidad 24](#_Toc18070548)

[**Vista de diseño** 25](#_Toc18070549)

[**Vista de despliegue (distribución de componentes)** 26](#_Toc18070550)

[**Vista de despliegue (Hardware)** 27](#_Toc18070551)

[**** 28](#_Toc18070552)

# **CONSIGNA**

# **ENUNCIADO**

# **E.R.S.**

# **ANÁLISIS**

## **Realización de caso de uso de análisis. Caso de uso 17 “Registrar pedido”**

## **Realización de caso de uso de análisis. Caso de uso 39 “Generar remito”**

## **Realización de caso de uso de análisis. Caso de uso 85 “Generar informe de efectividad”**

## 

## **Máquina de estados. Clase “Remito”**

# **ENTREGAS CON CORRECCIONES APLICADAS**

# **MODELO DE DOMINIO**

# **ARQUITECTURA**

## **Listado y descripción de Requerimientos No Funcionales**

## **Patrones arquitectónicos**

### N TIER

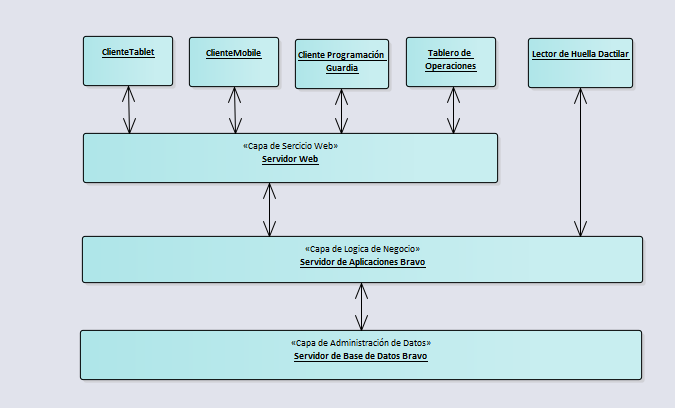
* Separación de intereses
* Comunicaciones sincrónicas
* Despliegue flexible

Motivación: La comunicación sincrónica entre capas brinda una mejor performance y confiabilidad en las transacciones.

La separación de intereses en distintas capas lógicas facilita la futura modificación y extensibilidad del sistema.

Nos permite utilizar clientes web delegados que no requieren demasiado hardware y también la distribución de la capa de datos en distintos servidores facilitando la redundancia ante eventuales catástrofes.

Aplicación: Corresponde a la vista de ejecución runtime que destaca sobre una estructura de capas, las comunicaciones entre las mismas y muestra la distribución de las capas en los niveles de hardware de la arquitectura.

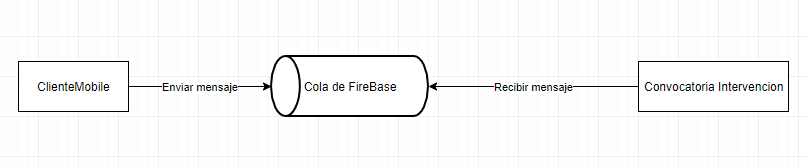


### MESSAGING

* Comunicación asincrónica
* Calidad de servicio configurable
* Bajo acoplamiento

Motivación: A medida que el sistema solicita el envío de mensajes, se enviaran los mensajes lo más rápidamente posible en un caso critico como es la notificación a bomberos de llamado a intervenciones. Y se asegurara que llegue en caso de no estar disponible, el mensaje quedara en la cola hasta que sea recibido.

Aplicación: Para resolver él envió de mensajes vía whatsapp a un grupo de destinatarios aprovechando la potencialidad del patrón y las comunicaciones asincrónicas que soporta la falta eventual de comunicación. También se utiliza al implementar un servicio de Google firebase que maneja las colas de mensajes que se enviaran como notificaciones hasta que los mismos sean transmitidos

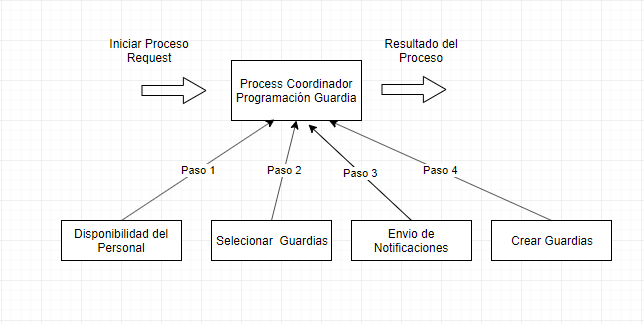


PROCESS COORDINATOR:

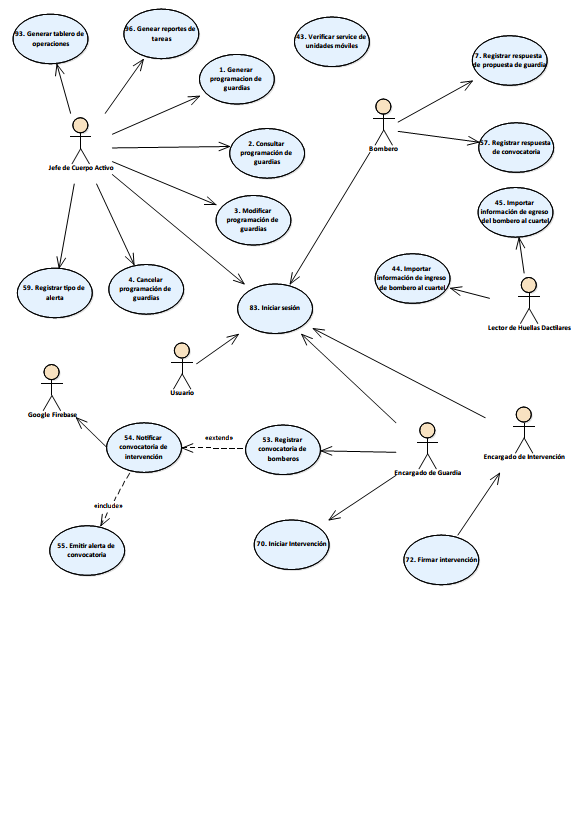
* Encapsulamiento del proceso
* Bajo acoplamiento
* Comunicación flexible

Motivación: Permite dividir un proceso en subprocesos que no se conocen entre sí sino que resuelven lo que les compete únicamente. Podremos así distribuir la carga en uno o más servidores y en caso de ser necesario modificar el código será mucho más fácil

Aplicación: Dividiremos el proceso de programación de guardias en subprocesos tales como: Disponibilidad del personal, edición de guardias, envío de notificaciones y creación de guardias con bomberos. Este patrón nos permite realizar este proceso independientemente de donde se encuentra distribuido cada componente.



## **Vista de la funcionalidad y justificación de casos de uso**

****

# Vista de la funcionalidad

|  |  |
| --- | --- |
| Caso de Uso | Justificación |
| 1. Generar programación de guardias, 2, 3, 4 | Es la transacción más compleja y además cumple con todos los RNF relacionados al desarrollo web. |
| 7. Registrar respuesta de propuesta de guardia | Relaciona al RNF 10, haciendo el uso de la Api de Google fireBase, que implementa el patrón messagin se soluciona el envió de notificaciones. |
| 43. Verificar service de unidades móviles | El cual corre una tarea programada para responder al RNF 13. |
| 44. Importar información de ingreso bombero al cuartel, 45 | Es el encargado del procesamiento de la huella dactilar y se relaciona al RNF 12. |
| 54. Notificar convocatoria de intervención | Relacionado al RNF 11, utilizando la API de Google FireBase |
| 55. Emitir alerta de convocatoria | Relacionado al RNF 11, utilizando la API de Google FireBase |
| 57. Registrar respuesta de convocatoria | Relacionado al RNF 9, que soluciona el envió y respuestas vía WhatsApp. |
| 59. Registrar Tipo alerta, 60, 61, 62 | Relacionado al RNF 19, soluciona el tipo de alerta. Notificación sonora convocatoria. |
| 72. Firma intervención | Relacionado al RNF 21, firma digital. |
| 83. Iniciar sesión, 84 | Relacionado al RNF 15, interfaz de acceso personalizado. |
| 93. Generar tablero de operaciones | Relacionado al RNF 17 y RNF 14, que hace que el tablero se refresque automáticamente. |
| 96. Generar reportes de tareas, 94, 95, 97, 98 | Relacionado al RNF 16, para la generación de reportes e informes. |

## **Vista de diseño**

## **Vista de despliegue (distribución de componentes)**

## **Vista de despliegue (Hardware)**

# 