Documento de análisis de requerimientos

|  |  |  |  |
| --- | --- | --- | --- |
| **Elaborado por:** | Andrea Manrique Pinilla | **Fecha de Elaboración:** | 10 marzo de 2018 |
| **Fecha de entrega** | 11 marzo de 2018 |

Control de cambios

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 10 marzo de 2017 | 1.0 | Análisis Requerimiento | Andrea Manrique Pinilla |

Tabla de contenido

[1. Descripción General 3](#_Toc508566160)

[1.2 Alcance 3](#_Toc508566161)

[2. Diseño Técnico 4](#_Toc508566162)

[3. Vista Lógica y de desarrollo 5](#_Toc508566163)

[3.1 Diagrama De paquetes 5](#_Toc508566164)

[3.2 Diagrama de Clases 5](#_Toc508566165)

[3.3 Diagrama de Secuencia 6](#_Toc508566166)

[3.4 Diagrama de Datos 8](#_Toc508566167)

[4. Vista Física 8](#_Toc508566168)

[4.1 Diagrama de Despliegue 8](#_Toc508566169)

[Monitoreo y Logs 9](#_Toc508566170)

# Descripción General

Existe un call center donde hay 3 tipos de empleados: operador, supervisor y director. El proceso de la atención de una llamada telefónica en primera instancia debe ser atendida por un operador, si no hay ninguno libre debe ser atendida por un supervisor, y de no haber tampoco supervisores libres debe ser atendida por un director.

# 1.2 Alcance

* Diseñar el modelado de clases y diagramas UML necesarios para documentar y comunicar el diseño.
* Debe existir una clase Dispatcher encargada de manejar las llamadas, y debe contener el método dispatchCall para que las asigne a los empleados disponibles.
* La clase Dispatcher debe tener la capacidad de poder procesar 10 llamadas al mismo tiempo (de modo concurrente).
* Cada llamada puede durar un tiempo aleatorio entre 5 y 10 segundos. Debe tener un test unitario donde lleguen 10 llamadas.

**EXTRAS**

* Dar alguna solución sobre qué pasa con una llamada cuando no hay ningún empleado libre.
* Dar alguna solución sobre qué pasa con una llamada cuando entran más de 10 llamadas concurrentes.
* Agregar los tests unitarios que se crean convenientes.
* Agregar documentación de código

NOTA : Se plante un paquete de vista para poder realizar una simulación desde un jsp para poder visualizar las llamadas , pero por tiempo no se alcanzó a implementar , la idea era generar un nodo para poder mostrar los empleados y ver como se gestionaba la llamada .

**Se considera fuera del alcance:**

* Cualquier funcionalidad no descrita en este documento.

# Diseño Técnico

El desarrollo se compone de un nuevo componente que me permita recibir 10 llamadas concurrentes con una clase Dispatcher encargada de manejar las llamadas, y debe contener el método dispatchCall para que las asigne a los empleados disponibles.

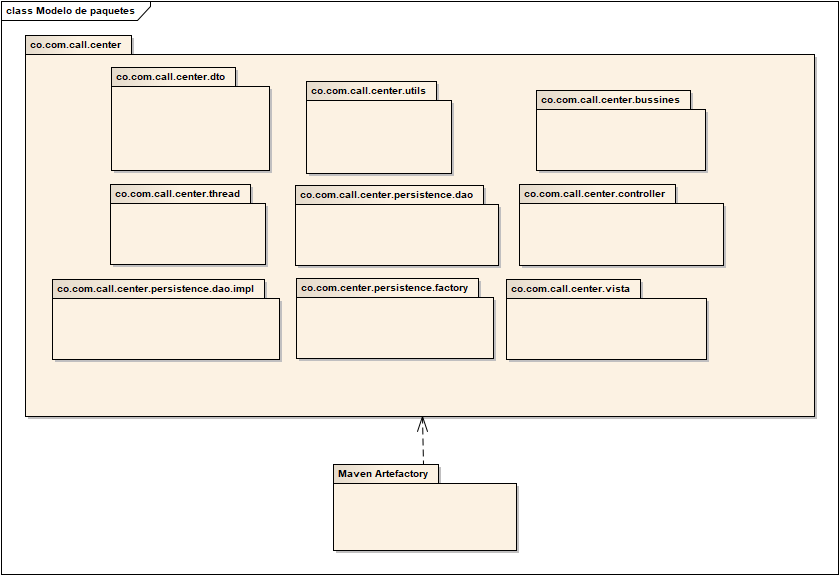
**Extras:**

Dar alguna solución sobre qué pasa con una llamada cuando no hay ningún empleado libre: para este punto se realizará una implementación de un servicio de devolución de llamada que consiste en registrar la información del cliente y cuando algún empleado se desocupe que realice la llamada **CU 04 .**

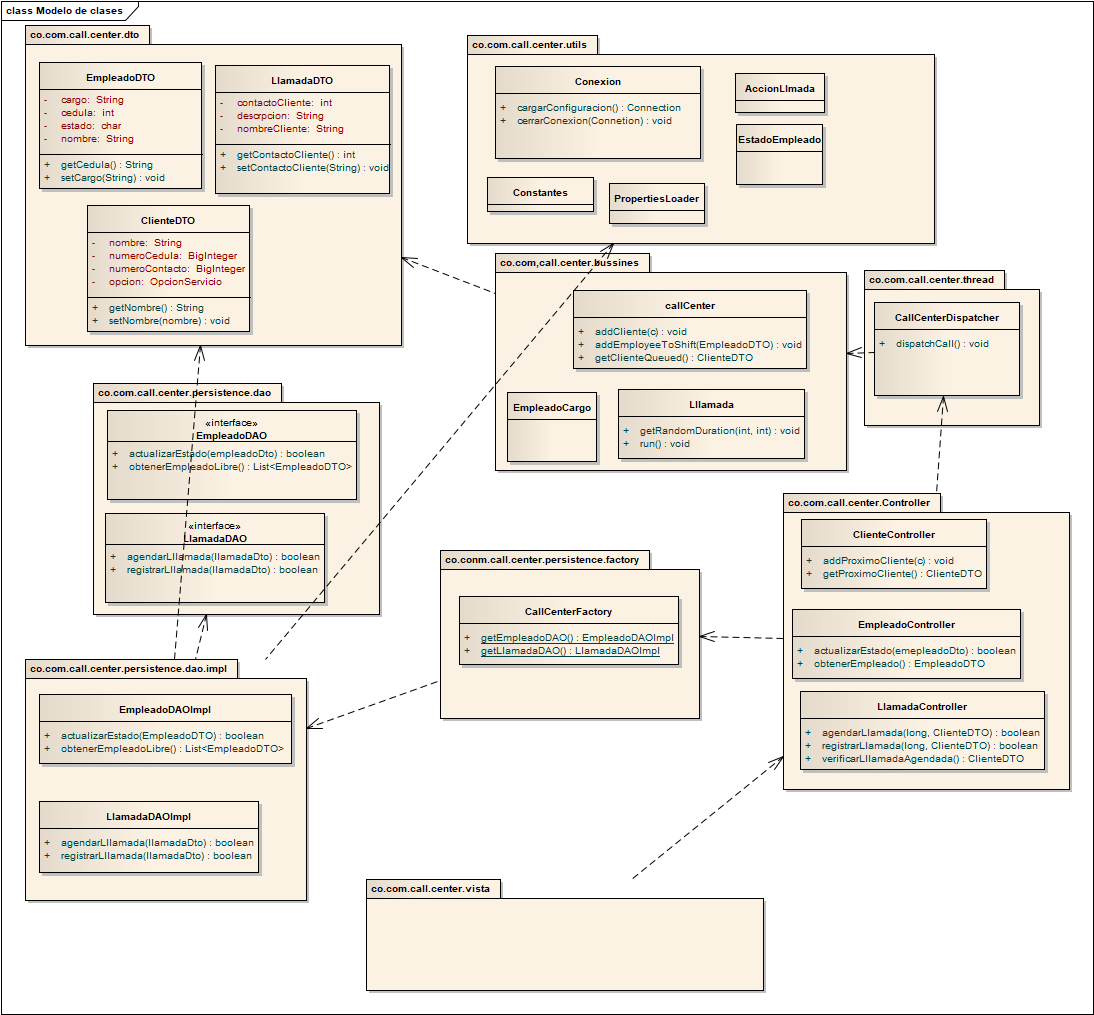
Dar alguna solución sobre qué pasa con una llamada cuando entran más de 10 llamadas concurrentes: para este punto se realizará una implementación de un pool de hilos en espera para ser atendidos, este corresponde al **CU 04 Espera en línea.**

# Vista Lógica y de desarrollo

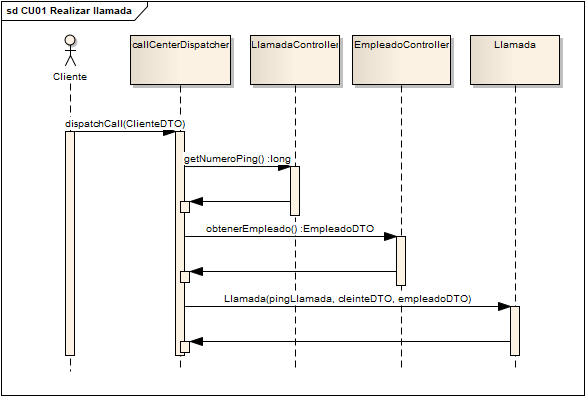
## Diagrama De paquetes

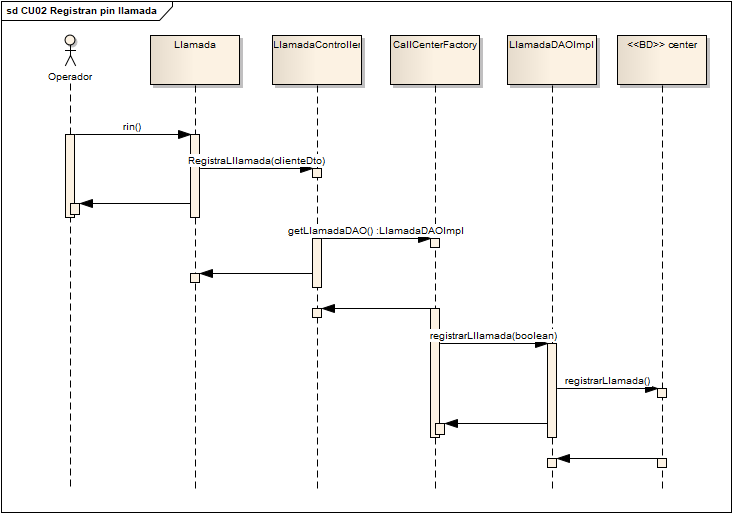


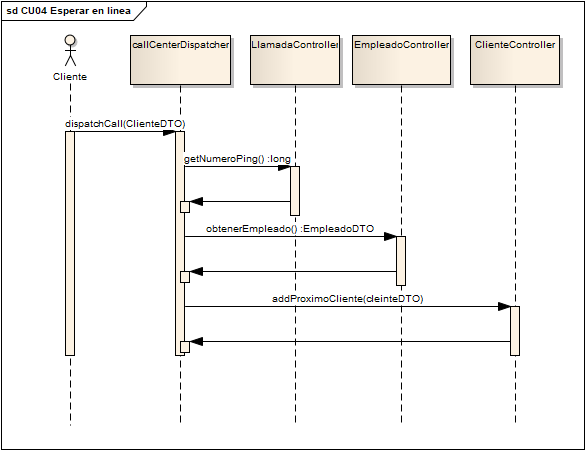
## Diagrama de Clases

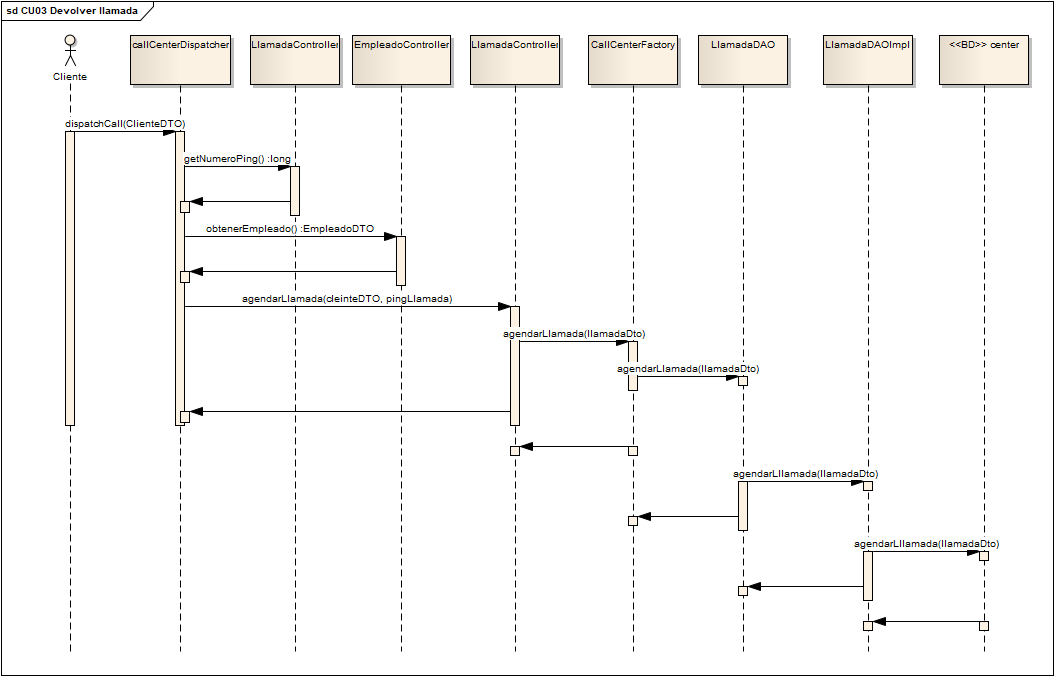


## Diagrama de Secuencia

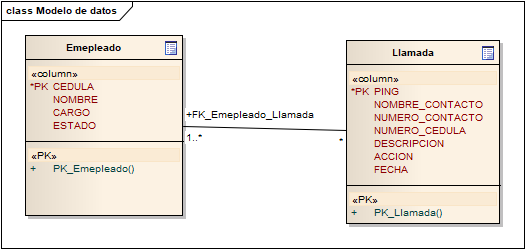






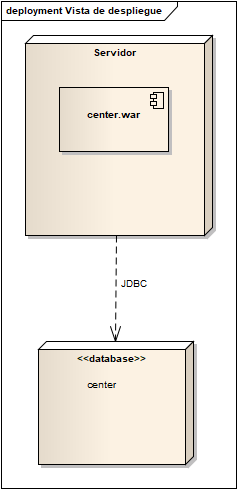


## Diagrama de Datos



# Vista Física

## Diagrama de Despliegue



Requerimientos No Funcionales

|  |  |  |
| --- | --- | --- |
| **Requerimiento no funcional** | **Detalle** | **Cómo cumplirlo** |
| Monitoreo y Logs | Se deben incluir logs de auditoria |  |