Proyecto Bases de Datos NOSQL.



Integrantes:

Lorena Romero - Código 67000230

Mauricio Rodríguez - Código 625467

Leonardo Oliveros - Código 624906

Presentado a:

Ingeniero Andrés Armando Sánchez Martín

Universidad Católica de Colombia

Facultad de Ingeniería

Programa de Ingeniería de Sistemas y Computación

Bases de Datos NOSQL

Bogota D.C.

2017 – 3

Contenido

[Introducción 3](#_30j0zll)

[Justificación 3](#_1fob9te)

[Objetivos 3](#_3znysh7)

[Objetivo General: 3](#_2et92p0)

[Objetivos Específicos 3](#_tyjcwt)

CorreoApp [3](#_3dy6vkm)

[Entorno 3](#_1t3h5sf)

[Descripción del problema 4](#_4d34og8)

[Solución planteada 4](#_2s8eyo1)

[Lista de Requerimientos de datos solicitados. 4](#_17dp8vu)

[Diseño conceptual de la estructura de los datos 4](#_3rdcrjn)

[Diseño lógico de los datos de la estructura de los datos 4](#_26in1rg)

Conclusiones y buenas prácticas

[Bibliografía y Referencias 4](#_lnxbz9)

# **Introducción**

La presente documentación es para dar a conocer al público en general que mediante esta nueva aplicación llamada CorreoApp se puede solucionar un problema identificado en agencias de paquetería.

Además, contará con la disponibilidad de las 24 horas del día, lo que beneficia dando la opción de realizar un monitoreo del estado del paquete que se ha enviado, es decir, un seguimiento a detalle en donde se encuentra actualmente el envío.

# **Justificación**

Como grupo hemos observado que el tiempo de entrega de paquetes no es el prometido y que las empresas responsables no cuentan con una app que informe el estado del paquete al cliente, y si lo tienen, se encuentra desactualizada. Esto genera insatisfacción al usuario y desprestigio a la empresa encargada por no cumplir con su función principal a tiempo. Lo que se busca es contar con una app que nos brinde el estado y los puntos que ha recorrido un paquete en específico.

Las ventajas que tendremos es aplicar lo aprendido a un caso de la vida real que necesita una atención pronta utilizando base de datos no relacionales. Se opta por una no relacional ya que nuestro sistema debe cumplir como prioridad la consistencia, de tal manera que los detalles y estados del paquete se encuentren actualizados en todos los nodos, y la tolerancia a particiones para que el sistema siga funcionando las 24 horas del día a pesar de que haya sido dividido por un fallo de comunicación (Teorema CAP). Se buscará una app óptima, con tiempo de respuesta bajos y con manejo de grandes volúmenes de datos de manera eficiente.

Se elige una base de datos orientado a clave valor ya que se caracterizan por tener un rendimiento alto, con alta curva de escabilidad y útil para representar datos no estructurados (Pérez, 2015). Hace más eficiente el manejo de grandes volúmenes de datos. Como motor se opta por Redis ya que se destaca por su velocidad; pero también por su sencillez de uso y flexibilidad.

# **Objetivos**

## Objetivo General:

Presentar una solución conforme a la trazabilidad de los envíos por un ente de servicio postal con el fin de brindar seguridad y confianza a los usuarios verificando el estado del paquete en un tiempo corto.

## Objetivos Específicos:

* Analizar la efectividad del sistema ante la sociedad.
* Establecer una propuesta para problemas de seguimiento de paquetes.
* Examinar cómo se comportan los datos y de qué manera se manejan mediante bases de datos no relacionales.
* Justificar por qué un modelo de seguimiento de paquetes sería una solución para pérdida o problemas de envío.

# **CorreoApp**

## **Entorno**

CorreoApp es una empresa dedicada a la prestación del servicio de paquetería por 10 años en el territorio Nacional Colombiano e Internacional. El servicio completo comprende la recolección, traslado, reparto y entrega en tiempos determinados. Se caracteriza por brindar seguridad y especialización de los servicios.

Sin embargo, en los últimos años, han tenido una disminución de clientela y esto ha sido porque otras empresas del mismo sector están implementando apps de seguimiento del paquete, lo cual es un atractivo muy grande para los usuarios de este servicio.

Por ende, el estado actual de la empresa no es muy favorable, ya que, si bien se han especializado por tantos años en el servicio, han dejado un lado la inversión a nuevas tecnologías. Es por ello, que se busca que su servicio contempla el rastreo y seguimiento de paquetes.

## **Descripción del problema**

El problema principal es que los clientes de este tipo de servicio buscan estar al día en cuanto al tiempo y lugar de sus paquetes (Ojanguren, 2017). El no estar al tanto de ello, puede ocasionarles incomodidad e inseguridad de que su paquete no se haya enviado o se haya quedado retenido en alguna parte del trayecto. Otro detalle, es que mucho de los asistentes de atención al cliente, no saben en qué parte del trayecto está el encargo pues el sistema no se encuentra actualizado.

Esto ha sido perjudicial para la empresa frente a otras. Correo App necesita invertir en métodos inteligentes con el fin de recuperar a sus clientes y su prestigio. Por otro lado, no estudian sus propios datos. No calculan métricas e indicadores que con una app se podrá tomar decisiones más oportunas

## **Solución planteada**

La solución es desarrollar una app que nos muestre a tiempo real los datos del paquete, con su respectivo estado, lugar, fecha y hora. Adicionalmente los datos del remitente y destinatario. Esta plataforma estará disponible las 24 horas del día lo que se adapta a cualquier horario. Esta app se desarrollará bajo una base de datos no relacional orientado a documento que utiliza como motor a MongoDB.

## **Lista de Requerimientos de datos solicitados.**

Para dar un servicio de tratamiento de información los ítems evaluados en nuestra propuesta son:

* Registrar usuario
* Ingresar al sistema
* Verificación del paquete
* Seguimiento del paquete
* Observaciones de paquete
* Confirmación de paquete o paquetes recibidos
* Obtener el estado del paquete
* Obtener detalles del paquete
* Registrar envio
* Modificar paquete
* Eliminar paquete
* Añadir estado de paquete

## **Descripción de Requerimientos**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nombre** | | R1 – Registrar usuario | | | | | |
| **Resumen** | | El sistema debe permitir dejar crear un usuario nuevo, para poder ingresar a la aplicación. | | | | | |
| **Entradas** | | | | | | | |
| Nombre, apellidos, correo electrónico, username y password. | | | | | | | |
| **Resultados** | | | | | | | |
| Nuevo usuario registrado en la BD | | | | | | | |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **Nombre** | | R2 – Ingresar al sistema | | | | | |
| **Resumen** | | El sistema debe permitir ingresar a la app con un usuario anteriormente registrado | | | | | |
| **Entradas** | | | | | | | |
| Username y password | | | | | | | |
| **Resultados** | | | | | | | |
| Ingreso al sistema | | | | | | | |
|  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Nombre** | R3 - Verificación del paquete |
| **Resumen** | El sistema verifica la existencia del código del paquete en el sistema |
| **Entradas** | |
| Código del paquete | |
| **Resultados** | |
| Paquete validado | |

|  |  |
| --- | --- |
| **Nombre** | R4 - Seguimiento del paquete |
| **Resumen** | El sistema muestra todos los puntos por donde el paquete ha pasado. |
| **Entradas** | |
| Código del paquete existente | |
| **Resultados** | |
| Registro de ubicaciones | |

|  |  |
| --- | --- |
| **Nombre** | R5 - Observaciones de paquete |
| **Resumen** | El sistema muestra al usuario todas las observaciones que el paquete tuvo antes del salir del local. Observaciones iniciales |
| **Entradas** | |
| Código del paquete existente | |
| **Resultados** | |
| Descripción de las observaciones | |

|  |  |
| --- | --- |
| **Nombre** | R6 - Confirmación de paquete o paquetes recibidos |
| **Resumen** | El sistema comunica al usuario que su paquete ha sido entregado y recibido en el domicilio destino. |
| **Entradas** | |
| Ninguna | |
| **Resultados** | |
| Paquete entregado | |

|  |  |
| --- | --- |
| **Nombre** | R7 - Obtener estado del paquete |
| **Resumen** | El sistema muestra al usuario el estado actual del paquete. |
| **Entradas** | |
| Código del paquete | |
| **Resultados** | |
| Ubicación del paquete | |

|  |  |
| --- | --- |
| **Nombre** | R8 – Obtener detalles del paquete |
| **Resumen** | El sistema muestra al usuario la descripción del paquete. Como el peso, el precio, el código, etc. |
| **Entradas** | |
| Código del paquete | |
| **Resultados** | |
| Descripción del paquete | |

|  |  |
| --- | --- |
| **Nombre** | R9 – Registrar paquete |
| **Resumen** | El sistema ingrese un nuevo paquete al sistema |
| **Entradas** | |
| Código de paquete inexistente, user y password de administrados | |
| **Resultados** | |
| Datos del paquete ingresado | |

|  |  |
| --- | --- |
| **Nombre** | R10 - Modificar paquete |
| **Resumen** | El sistema modifica un dato del paquete |
| **Entradas** | |
| Código del paquete existente, user y password de administrados | |
| **Resultados** | |
| Datos del paquete modificados | |

|  |  |
| --- | --- |
| **Nombre** | R11 – Eliminar paquete |
| **Resumen** | El sistema eliminar un envío en el sistema |
| **Entradas** | |
| Código del paquete existente, user y password de administrados | |
| **Resultados** | |
| Datos del paquete eliminado | |

|  |  |
| --- | --- |
| **Nombre** | R12 – Añadir estado de paquete |
| **Resumen** | El sistema añade la localización del paquete. |
| **Entradas** | |
| Código del paquete, username y password de usuario con permisos. | |
| **Resultados** | |
| Datos del paquete añadidos | |

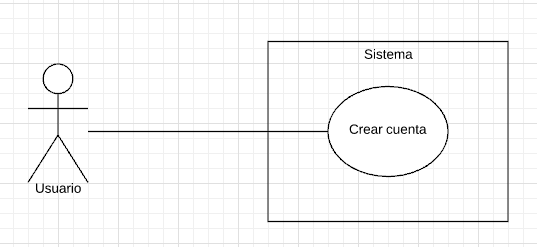
## **Diseño conceptual de la estructura de los datos**

Actores:

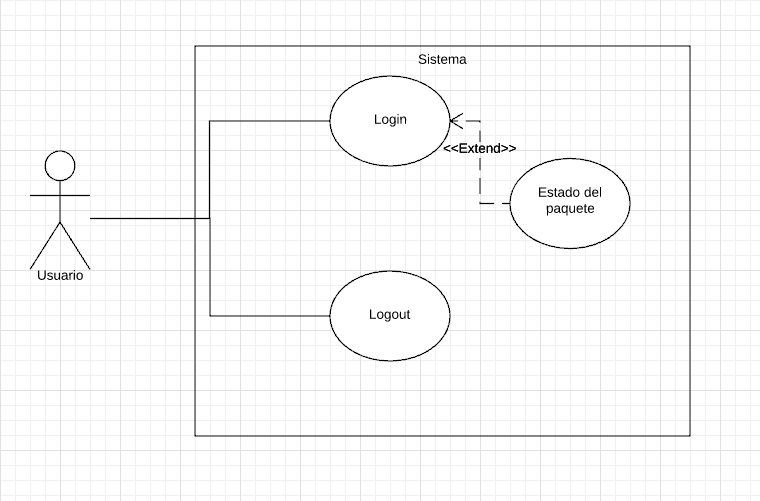
* Usuario
* Empleado
* Empresa

Casos de uso:

* Login de los usuarios



* Trazabilidad del paquete



* Seguimiento corporativo

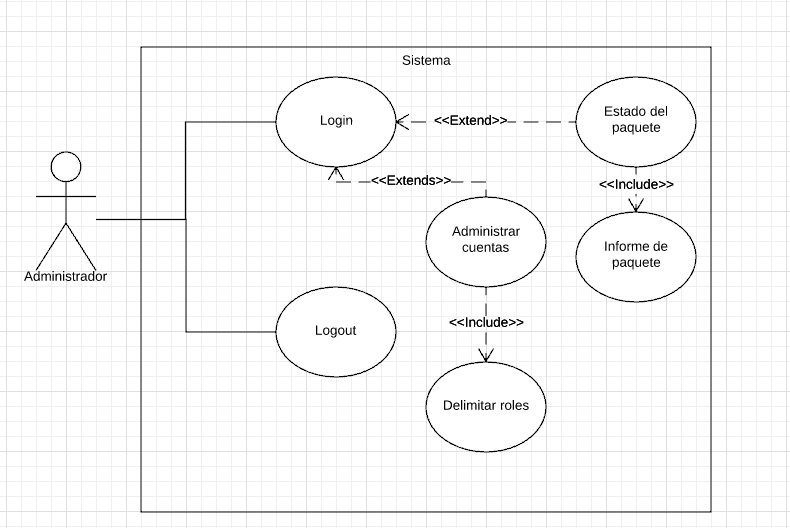


Diagrama de casos de uso de Negocio

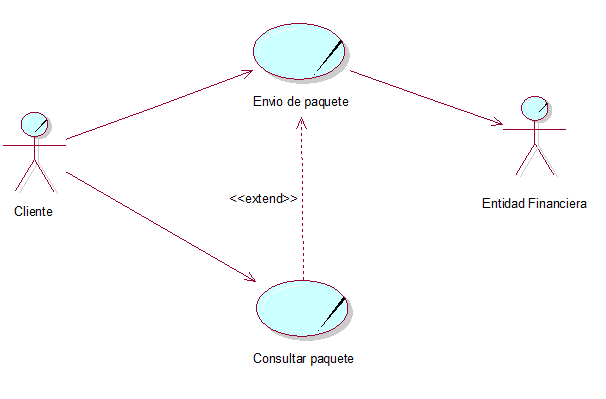
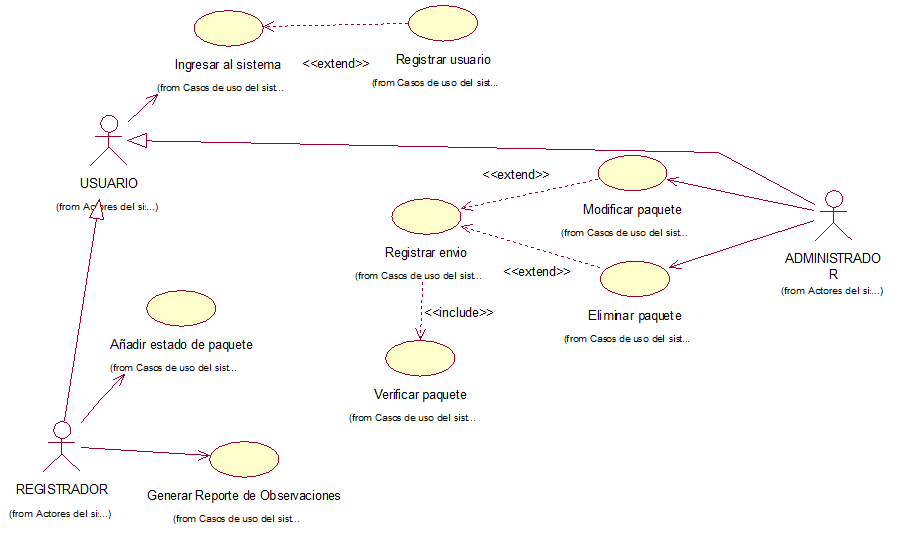


Diagrama de casos de uso del Sistema

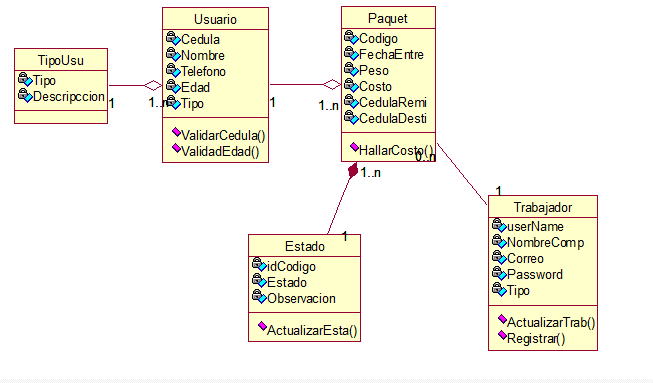
.

## **Casos de uso del sistema**

## **Diseño lógico de los datos de la estructura de los datos**

Diagrama de clases en estándar UML



Estructura de datos

Para la estructura de datos hemos considerado el uso de 2 hashes y 1 lista. Elegimos listas pues permiten almacenar secuencias de strings en el orden en el que fueron insertadas (recomendable para las inserciones del estado). Otra razón es porque las listas están optimizadas para que al añadir un elemento al inicio o al final de la lista tarde lo mismo independientemente de si la lista tiene 10 elementos, o 100 millones de elementos (Ayestarán, 2015). Para los datos del paquete y del usuario seleccionamos hash porque nos resulta más conveniente al momento de hacer consultas sobre un campo en específico. Por otro lado no acepta duplicados.

Hash 1: codigo NombresRemitente cedula edad teléfono NombreDestinatario cedula2 edad2 teléfono2 Fechaentrega peso origen destino costo FechaEstimada

Ejemplo: **10001** NombresRemitente  **“Ana Ramirez”** cedula  **22552** edad **45** teléfono **94265222** NombreDestinatario  **“Mateo Sánchez”** cedula2  **14555** edad2  **28** teléfono2 **98875233** Fechaentrega **“30/10/2017 13:34”** peso **2** origen **4** destino **1** costo  **180000** FechaEstimada **“2/11/2017 12:00”**

Lista2: codigo “estado” “Observación” “fecha”

Ejemplo: **estado:10001 “Simón bolívar” “Retenido por observación”**

Hash: username NombreCompleto correo password tipo

Ejemplo: **mateo123** NombreCompleto **“Juan Pérez”** correo **“jperes23@ucatolica.edu.co”** password **“Redis12”** tipo **1**

\*\*En el caso sea tipo 1 será Administrador, en el caso de tipo 2 será registrador.

CRUD’s

Para poder ejecutar los comandos CRUD’s consideremos las siguientes estructuras

|  |  |
| --- | --- |
| Sentencia | Utilidad |
| keys :\* | Trae todas las llaves o claves |
| exists CODIGO | Responde a la pregunta si el código ingresado existe. 1 si existe y si no 0. |
| lrange codigo 0 -1 | Muestra todos los elementos del codigo de inicio a fin. |
| rpush codigo | Agregar un nuevo elemento a la lista. |
| Lset codigo pos nuevodato | Modifica datos dado una posición (pos) y el nuevo valor a modificar (nuevodato). |
| del codigo | Eliminar la lista que tiene como clave dicho código |
| lindex codigo seq | Devuelve el valor del elemento que se encuentra en la posición (seq). |
| llen codigo | Devuelve la cantidad de elementos de la lista |
| lrem codigo cant dato | Elimina uno o varios elementos (cant) que contenga el dato en la lista. |

|  |  |
| --- | --- |
| Sentencia | Utilidad |
| keys :\* | Trae todas las llaves o claves |
| hvals CODIGO | Devuelve los valores de la clave. No devuelve los campos |
| Hset codigo campo valor | Modifica un nuevo valor al campo |
| HGet codigo campo | Trae el valor del campo dado un código |
| HMSET codigo campo "valor" campo1 "campo1” | Establece los campos especificados a sus respectivos valores |
| HMGET codigo | Devuelve los valores asociados del código. |
| DEL codigo CAMPO | Elimina el campo de un código. |

**Sentencias en terminal**

Estas pruebas se realizarán en la base de datos 2.

Create:

Caso listas

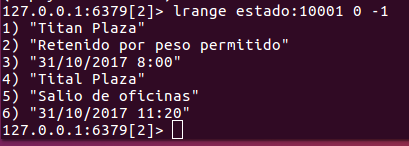


Caso hash

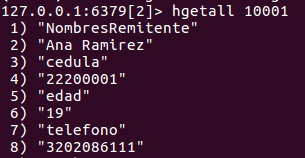


Read:

Caso listas:



Caso Hash:



Update:

Caso listas:

\*\* Se modifica a fecha del ultimo estado con la del 22/11/2017 a las 4:31



Caso hash:

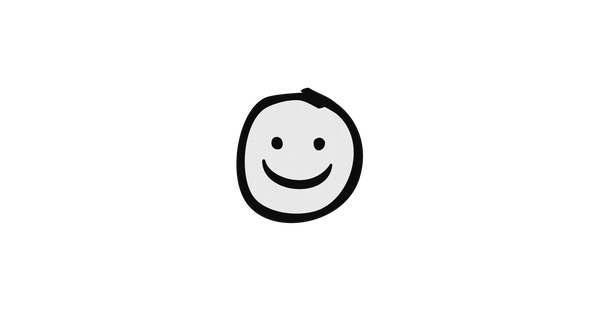
\*\*Se modificará la edad. De 45 años a 19 años



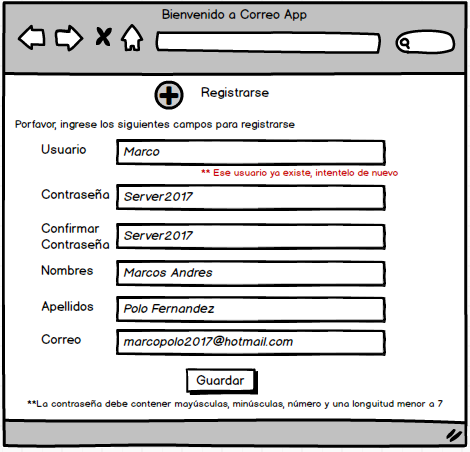
Delete:

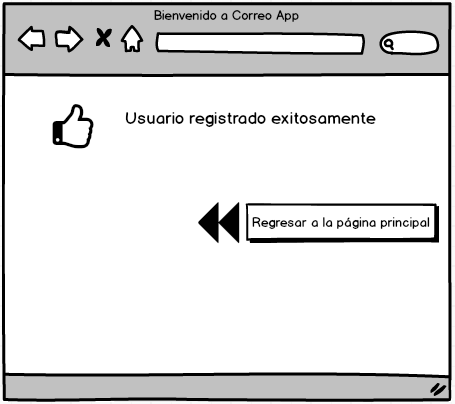


# **Representación gráfica**

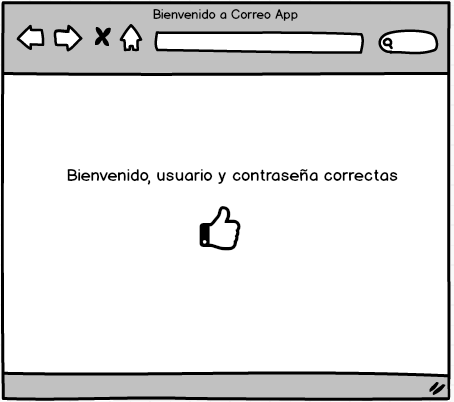


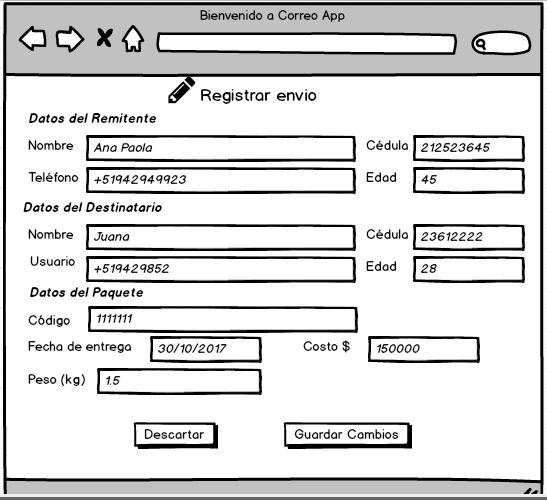
 

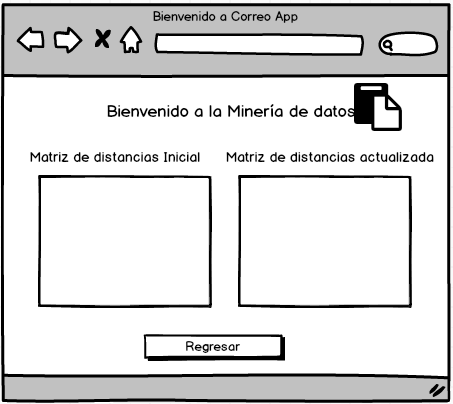


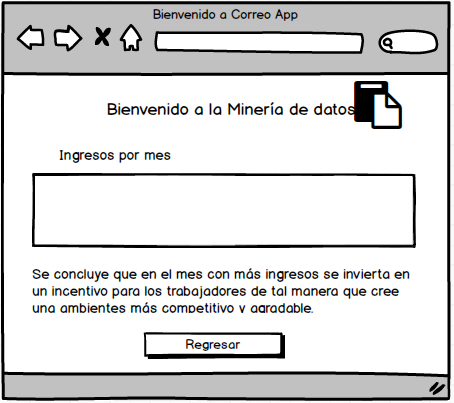
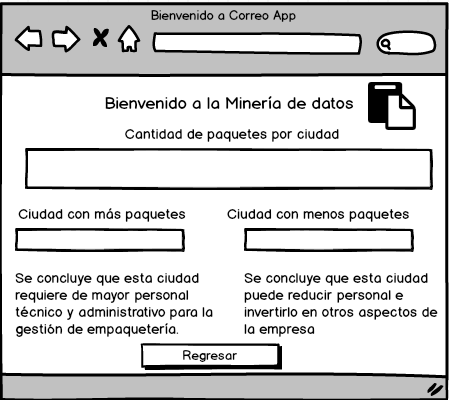


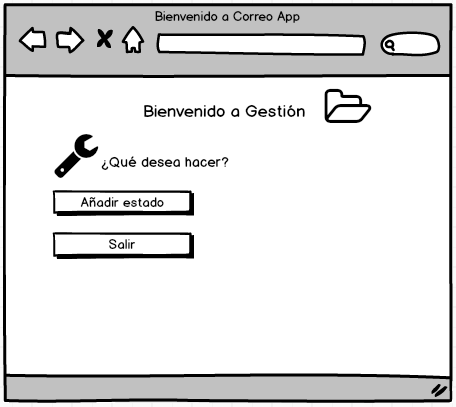


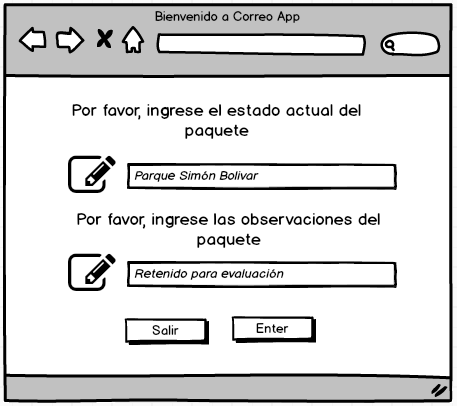


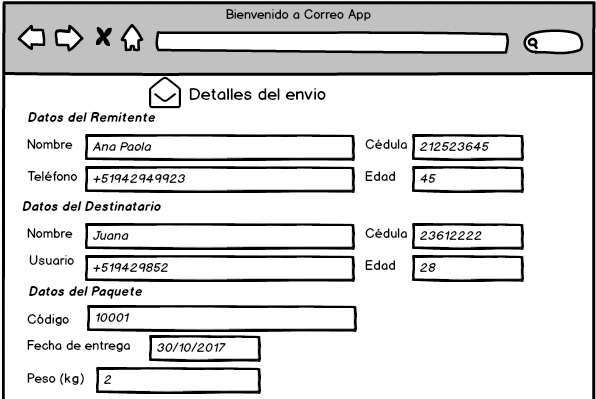


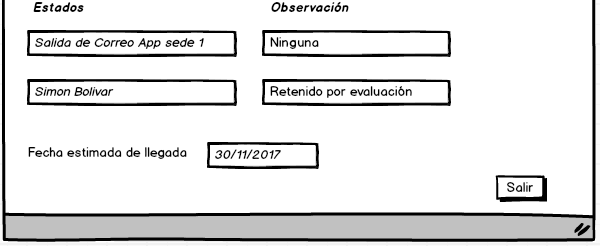












## **Pruebas de eficiencia**

Lo que busca esta aplicación es ofreceré al usuario final un mejor servicio, brindándole la trazabilidad de su paquete. De igual manera, contribuir a la toma de decisiones de la empresa, como la aproximación de días estimados de una ciudad a otro de tal manera que este lo más cercano al tiempo real. Otra decisión que podemos tomar de esta aplicación, es la contratación de personal en las localidades con más recepción de paquetes, como otras opciones de financiamiento y motivación.

# **Conclusiones y lecciones aprendidas**

A pesar de no estar muy familiarizamos con el sistema operativo Ubuntu, consideramos que el motor Redis es sumamente amigable al usuario, se aprende rápido y sus tiempos de respuesta son rápidas. Como reto, tenemos el desarrollo de la interfaz gráfica y la implementación de los nodos a nuestra aplicación. Nos resultó complicado pero no imposible de resolver.   
Por otro lado, este proyecto fortalece las habilidades del trabajo en equipo, la investigación y las buenas prácticas en cuanto a la eficiencia de la lógica al momento de programar.

Redis presenta diferentes tipos de datos y es importante analizar la data de que vamos a manejar y tomar la estructura que más de amolde a nuestros datos y a su consulta.

Concluimos que es sumamente importante la minería de datos. No solo se busca la funcionalidad de la aplicación, si no también datos cuantitativos y cualitativos para contribuir a la toma de decisiones de la empresa.

# Bibliografía

Ayestarán, V. (11 de Noviembre de 2015). *Paradigma*. Obtenido de Estructuras de datos en Redis: https://www.paradigmadigital.com/dev/no-solo-clave-valor-estructuras-de-datos-en-redis/

Pérez, S. (30 de Marzo de 2015). *Desarrollo web*. Obtenido de Desarrollo web: https://www.ondho.com/claves-para-elegir-tu-base-de-datos-nosql/

Strapappazzon, N. (12 de Junio de 2016). *Swapbytes*. Obtenido de Swapbytes: https://www.swapbytes.com/teorema-cap-base-datos/