# **CAPITULO 1: INTRODUCCION**

## **INTRODUCCION**

En la actualidad para el desarrollo de sistemas está centrado en la movilidad y acceso desde cualquier parte del mundo. Es decir tenga portabilidad y eficiencia para poder lograr un producto de calidad en el cual se pueda tener la información disponible sin necesidad de estar presentes en las oficinas correspondientes.

## **ANTECEDENTES**

La empresa de servicios fotográficos STUDIO S.A. fundada hace más de 3 años dedicada al servicio toma de fotografía e impresión.

El trabajo que realiza es el de prestar servicios en los eventos sociales, además la empresa tiene un servicio de venta de las fotografías, este tipo de ventas se lo realiza de dos maneras: mediante la impresión en papel o archivo digital.

## **DESCRIPCION DEL PROBLEMA**

### **SITUACION PROBLEMÁTICA**

La empresa dedicada al servicio de fotográfico cuenta con clientes y fotógrafos que realizan el servicio de toma de fotografías en los distintos eventos sociales. Pero al no poder conocer la identidad de cada persona que hay en cada una de las fotografías, no se puede intentar vender la fotografía con el consiguiente perdida de un posible comprador.

Además es necesario mostrar a los clientes del evento social donde estarán sus fotografías para que ellos puedan acceder sin necesidad de ir hasta la ubicación física de la empresa.

### **SITUACION DESEADA**

La empresa requiere una aplicación web que les permita gestionar un evento social, registrar clientes para que cada cliente pueda subir sus propias fotografías .

Con las fotografías del cliente el sistema mediante reconocimiento facial tendrá que mostrar solo las fotografías donde él aparece.

Además que para cada evento social debe de generar un código QR con la información especificada.

Tambien el sistema debe de contar con un sistema de envio de notificación al cliente, cada vez que aparezca en una nueva fotografía.

Tambien el sistema debe de contar con una galería de fotografías donde el cliente podrá visualizar las fotografías donde el aparece.

La fotografía que no haya comprada aparecerá con una marca de agua.

Ademas deberá de contar con un sistema para la adquisición de la fotografía mediante un sistema de pago online.

## **OBJETIVOS**

### **OBJETIVO GENERAL**

El objetivo es desarrollar un sistema que cumpla las funciones de automatizar las compras de fotografías y notificaciones a posibles compradores además de poder gestionar el perfil de cada usuario (fotógrafo o cliente).

### **OBJETIVOS ESPECIFICOS**

A continuación se describen los pasos a seguir para alcanzar el objetivo deseado, siguiendo los pasos que sugiere el Proceso Unificado de Desarrollo de Software (PUDS):

* Recopilar la información necesaria y suficiente acerca de las actividades que se realiza en UML y otras.
* Analizar toda la información recopilada para identificar y organizar los requisitos, logrando con esto una comprensión más precisa y adecuada de los mismos.
* Realizar el diseño del sistema en base a los requerimientos encontrados en el análisis, utilizando los modelos del Lenguaje Unificado de Modelado (UML) para representar gráficamente los modelos encontrados para el sistema.

## **METODOLOGIA DE DESARROLLO**

La metodología que se utilizara para el desarrollo del presente proyecto será El Proceso Unificado de Desarrollo de Software (PUDS), utilizando modelos y notación del Lenguaje Unificado de Modelado (UML) para representar gráficamente todos los modelos que se requerirán para todo el proceso del desarrollo.

### **FASES DE DESARROLLO**

Las fases del Proceso Unificado son el intervalo de tiempo entre dos hitos importantes del **proceso** durante el cual se cumple un conjunto bien definido de objetivos, se completan artefactos y se toman las decisiones sobre si pasar a la siguiente fase.

* Fase de Inicio.
* Fase de Elaboración.
* Fase de Construcción.
* Fase de Transición.

**Fases del proceso unificado**

Una fase está compuesta por iteraciones (versiones), cada iteración es un mini proyecto que resulta en un incremento del producto final.

Cada fase termina con hito, el cual se determina por la disponibilidad de un conjunto de artefactos (modelos, documentos), permitiendo tomar la decisión de continuar con la siguiente fase.

* **Fase de inicio**

Se pone en marcha el proyecto, se desarrolla una descripción del producto final, se representa el análisis de negocio cuanto será el costo y cuanto traerá beneficio, se delimita el alcance del proyecto y se decide la factibilidad del proyecto.

* **Fase de Elaboración**

Se obtienen requerimientos más detallados, realizando un análisis y diseño de alto nivel para establecer la arquitectura base, y se crea el plan para la construcción.

* **Fase de construcción**

El objetivo primordial de esta fase es dejar listo un producto software en su versión inicial, a veces llamada “versión beta”. Se desarrolla iterativa e incrementalmente el producto a través de las etapas de análisis, diseño, implementación y prueba, para ser entregado al usuario

* **Fase de transición**

El objetivo de esta fase es dejar satisfacción en todos los usuarios del software, implantando el producto en su entorno de operación, corrigiendo defectos detectados y efectuando pruebas de aceptación. Al concluir esta fase se decide si los objetivos han sido cumplidos, y si se debe iniciar otro ciclo de desarrollo. Esta fase finaliza con el lanzamiento del producto.

**Flujos de Trabajo Fundamental**

Un flujo de trabajo muestra todas las actividades que se pueden hacer para producir un conjunto particular de artefactos.

El PUDS cuenta de cinco flujos de trabajo principales:

* Requerimientos
* Análisis
* Diseño
* Implementación
* Pruebas

**Los flujos de trabajo fundamentales:**

A continuación se presentan los flujos de trabajo fundamentales del proceso unificado de desarrollo de software.

* **Captura de Requisitos**

El propósito de la captura de requisitos es ayudar al desarrollo del sistema correcto, permite a desarrolladores y clientes ponerse de acuerdo en esa descripción, como el cliente/usuario no es especialista en ordenadores se describe utilizando el lenguaje del cliente.

El trabajo de los requisitos se hace fundamentalmente durante el inicio y la elaboración.

* **Análisis**

El Análisis proporciona una visión general del sistema, se estudia la descripción de requisitos obtenidos en el flujo de trabajo anterior, refinándolos y estructurándolos con el propósito de alcanzar un mejor entendimiento de los mismos y obtener una descripción de requisitos del sistema centrándose en aspectos como la flexibilidad ante cambios en los requisitos (fácilmente entendible, mantenible) y reusabilidad cuando se construyan sistemas parecidos, el análisis se describe utilizando UML.

* **Diseño**

El propósito del diseño es encontrar la forma del sistema que cumpla con los requisitos, teniendo como entrada los resultados del análisis, que es expandido a una solución técnica.

Nuevas clases son agregadas para proporcionar una infraestructura técnica: las interfaces de usuario, el manejo de bases de datos para almacenar objetos en una base de datos. La estructura que impone el modelo de análisis deberá conservarse. El diseño es el centro de atención al final de la elaboración y el comienzo de las iteraciones de construcción.

* **Implementación**

Se implementa el software requerido por el cliente, las clases de la fase de diseño son convertidas a código actual en un lenguaje de programación orientado a objetos.

Se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. El propósito de la implementación es el desarrollar la arquitectura y el sistema como un todo.

La implementación abarca principalmente la fase de construcción continua durante la fase de transición para tratar defectos tardíos.

* **Prueba**

En esta fase se verifica el sistema entero, se realizan pruebas a cada resultado de la implementación para corregir la existencia de errores, al final de la prueba el sistema puede ser entregado al cliente.

Se verifican los resultados la implementación, probando cada construcción, incluyendo tanto construcciones internas como intermedias, así como las versiones finales a ser entregadas a terceros.

Las pruebas se centran principalmente en las fases de elaboración cuando se inicia la base ejecutable, construcción cuando el sistema esta implementado y transición para la corrección los defectos finales.

## **ALCANCE**

### **REQUISITOS FUNCIONALES**

* El sistema debe de tener la capacidad de gestionar perfil de usuario en el cual el usuario podrá subir sus fotos de perfil.
* La aplicación web debe de contar con un modulo de gestión de eventos sociales.
* Además el sistema debe de contar con un generador de código QR para guardar información del evento social.
* El sistema tendrá la capacidad de poder subir las fotografías de un evento social.
* La aplicación deberá de contar con un módulo para la generación de notificaciones que se envían a los clientes interesados en la adquisición de una fotografía.
* El sistema deberá de contar con una galería de imágenes donde se podrán apreciar las fotografías que no hayan sido adquiridas, con una marca de agua.
* La aplicación web deberá de contar con un mecanismo de autenticación.
* La aplicación deberá de contar con un modulo para la compra de fotografía mediante un sistema de pago en linea.
* El sistema deberá de contar con un modulo de reconocimiento facial para que se pueda identificar en que fotografías aparece el cliente, para asi poder enviar las notificaciones y poder visualizar en la galería de fotografías.

### **REQUISITOS NO FUNCIONALES**

* **Portabilidad:** El sistema puede ser usado en cualquier sistema operativo, indiferente de que plataforma use.
* **Funcionalidad:** El sistema cumple con las funciones descritas en la captura de requisitos. Además tiene la capacidad de entregar resultados correctos en base a los datos ingresados por el usuario.
* **Usabilidad:** El sistema tiene una bajo curva de aprendizaje, debido a que cuenta con la documentación manual de usuario donde viene una explicación de cómo se debe de utilizar el sistema.

## **TECNOLOGIA Y HERRAMIENTAS**

* La plataforma a emplear para el desarrollo de la aplicación web será en bajo PHP.
* El sistema de infraestructura a usar:
* **Windows** el sistema operativo a usar.
* **A**pache el servidor web de aplicación.
* **PostgreSql** el gestor de bases de datos.
* **P**hp como el lenguaje de programación.

# **CAPITULO 2: MARCO REFERENCIAL**

## **2.1. NOTACION UML**

**Introducción**

El lenguaje Unificado de Modelo (UML, Unified Modeling Lenguaje) es un lenguaje grafico para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software.

El objetivo de UML es describir cualquier tipo de sistema en términos de diagramas orientados a objetos.

**Modelo Conceptual de UML**

Para comprender UML, se necesitara adquirir un modelo conceptual del lenguaje y esto requiere aprender tres elementos principales: los bloques básicos de

Construcción, las reglas que indican como se pueden combinar estos bloques básicos y algunos mecanismos comunes que se aplican a través de UML.

**Bloques básicos de construcción de UML**

El vocabulario de UML incluye tres clases de bloques de construcción:

* Elementos.
* Relaciones.
* Diagramas.

### **ELEMENTOS EN UML**

Hay cuatro tipos de elementos en UML:

1.- Elementos Estructurales

2.- Elementos de Comportamiento

3.- Elementos de Agrupación

4.- Elementos de Anotación.

**1.- Elementos Estructurales.-** Los Elementos Estructurales son los nombres de los modelos de UML. En su mayoría son las partes estáticas de un modelo y representan cosas que son conceptuales o materiales.

**Clase.-** Una clase es una descripción de un grupo de objetos con propiedades (atributos), comportamiento (operaciones), relaciones a otros objetos, y semántica comunes. Por lo tanto, una clase es una plantilla para crear objetos.

Las clases usadas para describir sistemas y clasificar los objetos que identificamos en el mundo real. Gráficamente una clase se representa como un rectángulo, que normalmente incluye su nombre, atributos y operaciones.



**Interfaz.**-Es una colección de operaciones que especifican un servicio de una clase o componente. Por lo tanto, una interfaz describe el comportamiento visible externamente de ese elemento. Una interfaz define un conjunto de especificaciones de operaciones, pero nunca un conjunto de implementaciones de operaciones, Gráficamente se la representa con un círculo con su nombre:



**Colaboración.-**Define una interacción y es una sociedad de roles y otros elementos que colaboran para proporcionar un comportamiento cooperativo mayor que la suma de los comportamientos de sus elementos. Gráficamente, se la representa como una elipse de borde discontinua, incluyendo su nombre.

Nombre

**Caso de uso.-** Es una descripción de un conjunto de secuencias de acciones que un sistema ejecuta y produce un resultado observadle para un actor en particular.

Especifica el comportamiento de un sistema o de una parte de este. Aplicable a Sistemas, subsistemas, clases e interfaces individuales.

Caso de Uso

**Componente.-** Es la parte física y reemplazable de un sistema que conforma un conjunto de interfaces y proporciona la interpretación de dicho conjunto.

Un Componente representa típicamente el empaquetamiento físico de diferentes elementos lógicos.

Nombre

**Nodo.-** Es un elemento físico que existe en un tiempo de ejecución que representa un recurso computacional, que por lo general dispone de algo de memoria y, con frecuencia capacidad de procesamiento.

Nombre

**2.- Elementos de Comportamientos.-** Los elementos de comportamiento son las partes dinámicas de los modelos UML. Representan comportamiento en tiempo y espacio.

**Interacción.-** Es un comportamiento que comprende un conjunto de mensajes intercambios entre un conjunto de objetos, dentro de un contexto particular, para alcanzar un propósito especifico. Gráficamente, un mensaje se muestra como una línea dirigida incluyendo el nombre.

**Máquina de Estados.**-Es un comportamiento que especifica las secuencias de estados por as que pasa un objeto o una interacción durante su vida en respuesta a eventos, junto con sus reacciones estos eventos.

Nombre

**3.- Elementos de Agrupación.-** Son las partes organizativas de los modelos UML. Estos son las cajas en las que se pueden descomponer un modelo.

**Paquetes.-** Es un mecanismo de propósito general para organizar elementos en grupo. Los elementos estructurales, los elementos de comportamiento e incluso otros elementos de agrupación pueden incluirse en un paquete.

Nombre

**4.- Elementos de Anotación.-** Son las partes explicativas de los modelos UML.

Son comentarios que se pueden aplicar para describir, clasificar, y hacer Observaciones se sobre cualquier elemento de un modelo.

Descripción de

Lo que se quiere comunicar.

### **RELACIONES EN UML**

Hay cuatro tipos de relaciones en UML

1. Dependencia

2. Asociación

3. Generalización

4. Realización

**1. Dependencia.-** Es una relación semántica entre dos elementos, en la cual un cambio en un elemento (elemento independiente), puede afectar a la semántica del otro elemento (elemento dependiente).

Es una relación de uso. Se coloca cuando se desea expresar que un elemento utiliza a otro

**2. Asociación.-** Es una relación estructural que describe un conjunto de enlaces, los cuales son conexiones entre objetos. Especifica que los objetos de un elemento se conectan a los objetos de otro.

+patrón +empleado

**3. Generalización.-** Es una relación de especialización / generalización en la cual los objetos del elemento especializado pueden sustituir a los objetos del elemento general. Es decir el hijo comparte la estructura del padre.

Gráficamente se la representa como una línea con una flecha vacía apuntando al padre.

**4.- Realización.-** Relación semántica entre clasificadores, este es el que especifica un contrato que otro clasificador garantiza que cumplirá. Se utiliza en ele contexto de interfaces y colaboraciones, también entre un caso de uso y su colaboración.

Gráficamente se la representa:

### **DIAGRAMAS UML**

Un diagrama es la representación gráfica de un conjunto de elementos, visualizados la mayoría de las veces como un grafo conexo de nodos (elementos) y arcos (relaciones).

Un diagrama representa una vista resumida de los elementos que constituyen en un sistema.

UML incluye nueve diagramas.

#### **Diagramas Estructurales**

**1.- Diagrama de clases.-** Se utilizan para describir la vista de diseño estática de un sistema, muestra un conjunto de clases, interfaces y colaboraciones. El diagrama de clase es el diagrama principal para el análisis y diseño.

**2.**-**Diagrama de objetos.-** Describen estructuras de datos, instantáneas de las instancias de los elementos encontrados en los diagramas de clases. (Objetos)

**3.-Diagramas de Componentes.-** Muestra la organización de la dependencia entre un conjunto de componentes. Se utilizan para describir la vista de implementación estática de un sistema. (Componentes).

Un componente es un grupo de clases que trabajan estrechamente. Los componentes pueden corresponder código fuente, binario o ejecutable.

**4.-Diagrama de Despliegue (Distribuido).-** Se utilizan para describir la vista de despliegue estática de una arquitectura. (Nodos).El diagrama de distribución modela la distribución en tiempo de ejecución de los elementos d procesamiento y componente software, junto a los procesos y objetos asociados. E n el diagrama de distribución se modela los nodos y la comunicación entre ellos.

#### **Diagramas de Comportamiento**

**1. Diagramas de Casos de Uso.-** Muestra un conjunto de casos de usos y actores y sus relaciones. Es una técnica para capturar información de cómo un sistema o negocio trabaja actualmente o de cómo se desea que trabaje, para el modelado de escenarios en los cuales el sistema debe operar.

**2.- Diagrama de Secuencia.-** Se centra en la organización temporal de los mensajes. Representa un conjunto de objeto y mensajes enviados y recibidos (Objetos, Instancias de Clases, Colaboraciones, Componentes y Nodos)

**3.- Diagrama de Colaboración.-** Se centra en la organización estructural de los objetos que envían y reciben mensajes. Muestra un conjunto de objetos, enlaces entre estos y mensajes enviados y recibidos por estos. (Los diagramas de Colaboración y Secuencia son Isomorfos, se pueden convertir de uno a otro sin pérdida de información).

**4.- Diagramas de Estados.-** Representa una máquina de estados, constituida por estados, transiciones, eventos y actividades. Se utilizan para describir la vista de un sistema. El diagrama de Estado modela el comportamiento de una parte del sistema a través del tiempo. Típicamente se elabora un diagrama de estado para cada clase que tenga un comportamiento significativo.

El comportamiento es modelado en términos del estado en el cual se encuentra el objeto, que acciones se ejecutan en cada estado y cual es el estado al que transita después de un determinado evento.

**5.- Diagrama de Actividades.-** Muestra el flujo de actividades de un sistema, conjunto de actividades, el flujo secuencial o ramificado de las actividades, y los objetos que actúan y sobre lo que se actúa.

Caso especial de Diagrama de Estados donde: todos (o la mayoría de) los estados son estados de acción, todas (La mayoría de) las transiciones son “Disparadas” como consecuencia de la finalización de la acción. El Diagrama puede estar asociado a: Una clase, la implementación de una operación o a un Caso de Uso.

# **CAPITULO 3: REQUISITOS**

## **MODELO DE DOMINIO**



## **IDENTIFICAR ACTORES Y CASOS DE USO**

### **IDENTIFICACION DE ACTORES**



**Usuario:** Eselusuario en el que se generalizan los demás usuarios específicos.

* **Fotógrafo:** Esel fotógrafo que se encargara de crear los eventos sociales y subir las fotos de los eventos sociales.
* **Cliente:** Es la persona que se registra en el sistema para poder consultar sus fotografías y poder realizar compras.

### **IDENTIFICACION DE CASOS DE USO**

En esta sección identificaremos los casos de uso del sistema de producción

CU01: Gestionar Perfil de usuario.

CU02: Gestionar Eventos sociales.

CU03: Gestionar Fotografías de perfil.

CU04: Gestión de fotografías de eventos sociales.

CU05: Galería de fotografías de usuario.

CU06: Gestionar compras de fotografías.

CU07: Modulo de reconocimiento facial.

### **PRIORIZACION DE CASOS DE USO**

#### **Criterios de priorización**

Para la siguiente priorización de los casos de usos se ha evaluado cada caso de uso contra cada uno de los siguientes criterios asignándoles valor entre 0 y 3

0 = no aplica

1 = poco relevante

2 = medianamente relevante

3 = muy relevante

Para cada caso de uso se ha sumado el valor obtenido en cada criterio:

1. El caso de uso tiene impacto importante en el diseño del sistema, puede ser porque agrega muchas clases a la capa intermedia o porque requiere muchos servicios de persistencia (capa de base de datos).
2. El análisis del caso de uso facilitará profundizar en el diseño del sistema a un costo de tiempo relativamente bajo.
3. El caso de uso involucra riesgos importantes asociados con tiempos de entrega, funciones complejas, tecnología poco conocida, personal con experiencia cuya presencia en el proyecto es inestable u otros.
4. El caso de uso representa un proceso muy importante del negocio.
5. El desarrollo del caso de uso representará ahorro en costos o ganancias significativas para la organización usuaria.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Caso de Uso** | **1** | **2** | **3** | **4** | **5** | **Total** |
| CU01 | 2 | 1 | 1 | 2 | 0 | 6 |
| CU02 | 2 | 1 | 1 | 2 | 0 | 6 |
| CU03 | 1 | 0 | 1 | 0 | 0 | 2 |
| CU04 | 1 | 0 | 1 | 0 | 0 | 2 |
| CU05 | 2 | 2 | 1 | 2 | 1 | 8 |
| CU06 | 3 | 2 | 1 | 1 | 0 | 7 |
| CU07 | 3 | 2 | 1 | 1 | 0 | 7 |

#### **Priorización**

|  |  |  |
| --- | --- | --- |
| CU01 | Gestionar perfil de usuario | MEDIA |
| CU02 | Gestionar eventos sociales | MEDIA |
| CU03 | Gestionar fotografías de perfil. | MEDIA |
| CU04 | Gestionar fotografías de eventos sociales | MEDIA |
| CU05 | Galería de fotografías de usuario. | BAJA |
| CU06 | Gestionar compras de fotografías | BAJA |
| CU07 | Módulo de reconocimiento facial | ALTA |

## **DETALLAR CASOS DE USO**

### **3.3.1. CU01: Gestionar perfil de usuario.**

#### **3.3.1.1. Diagrama de caso de uso**



#### **3.3.1.2. Descripción del caso de uso**

|  |  |  |  |
| --- | --- | --- | --- |
| **CASO DE USO: Gestionar perfil de usuario** | | | |
| **ID :** | **CU01.** | | |
| **Creado por:** | Alex Limbert Yalusqui Godoy | **Última modificación por:** | Alex Limbert Yalusqui Godoy |
| **Fecha de creación:** | 21-enero-2017 | **Fecha de última modificación:** | 21-enero-2017 |
| **Actores Primarios:** | Usuario | | |
| **Actores secundarios:** |  | | |
| **Descripción:** | Permite crear un perfil de usuario donde introduce los datos personales del empleado como nombre completo, dirección, teléfono, correo electrónico, Nick de usuario y contraseña de perfil.  Además podemos modificar nuestro propio perfil de usuario. | | |
| **Precondiciones:** | * Para modificar perfil debe de estar autenticado correctamente. * Para crear nuevo perfil, no debe de existir una sesión iniciada. | | |
| **Postcondiciones:** | * Los cambios realizados se actualizarán en el sistema. | | |
| **Flujo Principal:** | 1. El caso de uso inicia cuando el usuario ingresa a la pagina web y hace clic en perfil de usuario. 2. El sistema desplegara un formulario para que el usuario introduza los datos requeridos. 3. Una vez introducido los datos correspondientes.    1. El usuario hace clic en el botón de “Guardar”-    2. El sistema validara los datos introducidos. 4. Si los datos son correctos se guardara en el servidor. 5. El sistema nos mostrara una vista de todos los datos previamente introducidos. | | |
| **Flujo(s)**  **alternativo(s):** | 1. En el punto 3.2, si los campos no se validan correctamente.    1. El sistema informa que campo o campos no cumplen con los requisitos necesarios para grabar los cambios. | | |

### **3.3.2. CU02: Gestionar eventos sociales.**

#### **3.3.2.1. Diagrama de caso de uso**



#### **3.3.2.2. Descripción del caso de uso**

|  |  |  |  |
| --- | --- | --- | --- |
| **CASO DE USO: Gestionar Eventos sociales** | | | |
| **ID :** | **CU02.** | | |
| **Creado por:** | Alex Limbert Yalusqui Godoy | **Última modificación por:** | Alex Limbert Yalusqui Godoy |
| **Fecha de creación:** | 22-enero-2015 | **Fecha de última modificación:** | 22-enero-2017 |
| **Actores Primarios:** | Fotografo | | |
| **Actores secundarios:** |  | | |
| **Descripción:** | Permite crear un evento social donde ingresamos los datos correspondiente del evento social, dirección, fecha de evento y datos para el codigo QR.  Además podemos modificar, eliminar, buscar un registro ya creado anteriormente. | | |
| **Precondiciones:** | * El fotógrafo debe de estar autenticado correctamente en el sistema. | | |
| **Postcondiciones:** |  | | |
| **Flujo Principal:** | 1. El caso de uso inicia cuando el fotografo elige en el menú de la interfaz de usuario “Eventos Sociales”. 2. El sistema mostrará la interfaz que consta de una lista paginada de los Eventos sociales disponibles, mostrando los campos: descripción del evento, dirección, fecha de realización y teléfono de contacto. con las opciones de **subir fotos**, **modificar**, **eliminar** por cada evento social en la lista desplegada. Además despliega las opciones de búsqueda avanzada y nuevo Evento social. 3. Si el fotógrafo elige una de las opciones del evento social:    1. Si el fotografo elige la opción de **modificar** un evento.       1. El sistema mostrará un formulario de edición de datos del evento social mostrando los campos: descripción, direccion, fecha, teléfono e información para el código QR.   Además las opcion de Actualizar.   * + 1. El fotografo edita cualquiera de los campos listados como editables.     2. El almacenero elige la opción de actualizar.     3. El sistema verifica la validez de los datos a grabar.     4. El sistema graba en la base de datos los datos modificados e informa al fotografo que la operación se realizó exitosamente mediante un mensaje. Y visualizando en detalle todos los campos actualizados del evento social.   1. Si el fotografo elige la opción de eliminar evento social      1. El sistema pide confirmación al fotografo      2. Si el fotografo confirma la eliminación         1. El sistema elimina el evento social         2. El sistema visualiza el mensaje de que la eliminación fue exitosa.         3. El sistema actualiza sus datos de control interno.      3. Caso contrario         1. El sistema retorna a la vista de listado de eventos sociales.   2. Si el fotografo elige realizar una búsqueda avanzada   3.3.1. Introduce en el campo de búsqueda su criterio a buscar.  3.3.2. Se desplegará en la lista de eventos sociales solo aquellos que cumplan el criterio de búsqueda.   1. Si el fotografo elige Nuevo Evento social 2. El sistema visualiza la interfaz con un formulario para ingresar los datos del evento social como descripcion, direccion, fecha, teléfono y datos para el código QR.   Además de las opción de grabar y retornar   1. El fotografo ingresa todos los datos requeridos del evento social 2. Si el almacenero elige la opción de **actualizar**. 3. El sistema verifica la validez de los datos a grabar. 4. El sistema graba en la base de datos los datos ingresados e informa al fotógrafo que la operación se realizó exitosamente. Y visualizando en detalle todos campos ingresados del evento social. | | |
| **Flujo(s) alternativo(s):** | 1. En el punto 2. del flujo principal, si no existe ningún evento social registrado.    1. El sistema visualizará solo las entradas de búsqueda, pero no visualizará ningún evento social. 2. En el punto 3.1.4, si los campos no cumplen la validación.    1. El sistema informa que campo o campos no son válidos y no procede a grabar los cambios. 3. En el punto 4.4. si los campos no cumplen alguna política de negocio. 4. El sistema informa que campo o campos están vacíos y no procede a grabar los cambios. | | |

### **3.3.3. CU03: Gestionar fotografía de perfil.**

#### **3.3.3.1. Diagrama de caso de uso**



#### **3.3.3.2. Descripción del caso de uso**

|  |  |  |  |
| --- | --- | --- | --- |
| **CASO DE USO: Gestionar fotografía de perfil** | | | |
| **ID :** | **CU03.** | | |
| **Creado por:** | Alex Limbert Yalusqui Godoy | **Última modificación por:** | Alex Limbert Yalusqui Godoy |
| **Fecha de creación:** | 22-enero-2017 | **Fecha de última modificación:** | 22-enero-2017 |
| **Actores Primarios:** | cliente | | |
| **Actores secundarios:** |  | | |
| **Descripción:** | Permite subir, eliminar fotografía del perfil en el servidor. | | |
| **Precondiciones:** | * El cliente debe de estar autenticado en el sistema. | | |
| **Postcondiciones:** | * Las fotografías subidas se utilizaran para el reconocimiento facial y realizar las búsquedas dentro de las fotografías de eventos sociales. | | |
| **Flujo Principal:** | 1. El caso de uso inicia cuando el cliente elige en el menú de la interfaz de usuario “Detección facial”. 2. El sistema mostrará la interfaz que consta de una lista paginada de los las fotografía de perfil disponible, mostrando los campos: nombre de fotografía y fecha de subida. con las opciones de modificar, eliminar, visualizar por cada evento social en la lista desplegada. Además despliega las opciones de búsqueda avanzada y “**Subir fotografía**”. 3. Si el cliente elige una de las opciones del evento social:    1. Si el cliente elige la opción de modificar un evento social       1. Opción no habilitada.    2. Si el cliente elige la opción de eliminar fotografia       1. El sistema pide confirmación al cliente       2. Si el cliente confirma la eliminación          1. El sistema elimina la fotografía del servidor.          2. El sistema visualiza el mensaje de que la eliminación fue exitosa.    3. Si el cliente elige realizar una búsqueda avanzada   3.3.1. Introduce en el campo de búsqueda su criterio a buscar.  3.3.2. Se desplegará en la lista de fotografias solo aquellos que cumplan el criterio de búsqueda.   1. Si el cliente elige “**Subir fotografía**” 2. El sistema muestra un formulario para subir una fotografía. 3. Si el cliente hace clic en “**seleccionar fotografia**” y Subir. 4. El sistema verifica la validez de los datos a subir. 5. El sistema graba en el servidor la fotografía Y visualiza la imagen. | | |
| **Flujo(s) alternativo(s):** | 1. En el punto 4.3 si la fotografía no cumple los requisitos. Se mostrara un mensaje al cliente. | | |

### **3.3.4. CU04: Gestionar fotografías de eventos sociales.**

#### **3.3.4.1. Diagrama de caso de uso**



#### **3.3.4.2. Descripción del caso de uso**

|  |  |  |  |
| --- | --- | --- | --- |
| **CASO DE USO: Gestionar fotografías de eventos sociales** | | | |
| **ID :** | **CU04.** | | |
| **Creado por:** | Alex Limbert Yalusqui Godoy | **Última modificación por:** | Alex Limbert Yalusqui Godoy |
| **Fecha de creación:** | 22-enero-2017 | **Fecha de última modificación:** | 22-enero-2017 |
| **Actores Primarios:** | Fotografo | | |
| **Actores secundarios:** |  | | |
| **Descripción:** | Permite subir, eliminar fotografía de los eventos sociales al servidor. | | |
| **Precondiciones:** | * El fotografo debe de estar autenticado en el sistema. | | |
| **Postcondiciones:** | * Las fotografías subidas se utilizaran para el reconocimiento facial. | | |
| **Flujo Principal:** | 1. El caso de uso inicia cuando el fotografo elige dentro de la lista de eventos sociales “Subir Fotografias de evento” 2. El sistema mostrará la interfaz que consta de una lista paginada de los las fotografía del evento social disponibles, mostrando los campos: nombre de fotografía y fecha de subida. con las opciones de modificar, eliminar, visualizar por cada fotografia en la lista desplegada. Además despliega las opciones de búsqueda avanzada y “**Subir fotografía**”. 3. Si el cliente elige una de las opciones del evento social:    1. El cliente elige la opción de modificar fotografia       1. Opción no habilitada.    2. El cliente elige la opción de eliminar fotografia       1. El sistema pide confirmación al cliente       2. Si el cliente confirma la eliminación          1. El sistema elimina la fotografía del servidor.          2. El sistema visualiza el mensaje de que la eliminación fue exitosa.    3. Si el cliente elige realizar una búsqueda avanzada   3.3.1. Introduce en el campo de búsqueda su criterio a buscar.  3.3.2. Se desplegará en la lista de fotografias solo aquellos que cumplan el criterio de búsqueda.   1. Si el cliente elige “**Subir fotografía**” 2. El sistema muestra un formulario para subir una fotografía. 3. Si el cliente hace clic en “**seleccionar fotografía**” y Subir. 4. El sistema verifica la validez de los datos a subir. 5. El sistema graba en el servidor la fotografía Y visualiza la imagen. | | |
| **Flujo(s) alternativo(s):** | 1. En el punto 4.3 si la fotografía no cumple los requisitos. Se mostrara un mensaje al cliente. | | |

### **3.3.5. CU05: Galería de fotografía de usuario.**

#### **3.3.5.1. Diagrama de caso de uso**



#### **3.3.5.2. Descripción del caso de uso**

|  |  |  |  |
| --- | --- | --- | --- |
| **CASO DE USO: Galeria de fotografia de usuario** | | | |
| **ID :** | **CU05.** | | |
| **Creado por:** | Alex Limbert Yalusqui Godoy | **Última modificación por:** | Alex Limbert Yalusqui Godoy |
| **Fecha de creación:** | 22-enero-2017 | **Fecha de última modificación:** | 22-enero-2017 |
| **Actores Primarios:** | Cliente | | |
| **Actores secundarios:** |  | | |
| **Descripción:** | Visualizará una galería de imagen donde se encontraran todas las fotografías en las cuales se encuentre el cliente. | | |
| **Precondiciones:** | * El cliente debe de estar autenticado correctamente en el sistema. * Debe haberse cumplido el caso de uso CU03. | | |
| **Postcondiciones:** |  | | |
| **Flujo Principal:** | 1. El caso de uso inicia cuando hace clic en el menú que dice “**Galeria**”. 2. Se mostrará una galería de imágenes con con marca de agua,   En todas donde el cliente aparece.   1. En cada fotografía tendra una opción para realizar la compra de la fotografía. | | |
| **Flujo(s) alternativo(s):** |  | | |

### **3.3.6. CU06: Gestionar compra de fotografia.**

#### **3.3.6.1. Diagrama de caso de uso**



#### **3.3.6.2. Descripción del caso de uso**

|  |  |  |  |
| --- | --- | --- | --- |
| **CASO DE USO: Gestionar compra de fotografía** | | | |
| **ID :** | **CU06.** | | |
| **Creado por:** | Alex Limbert Yalusqui Godoy | **Última modificación por:** | Alex Limbert Yalusqui Godoy |
| **Fecha de creación:** | 23-enero-2017 | **Fecha de última modificación:** | 23-enero-2017 |
| **Actores Primarios:** | Cliente | | |
| **Actores secundarios:** |  | | |
| **Descripción:** | Permite realizar una compra de fotografías previamente seleccionadas. | | |
| **Precondiciones:** | * El cliente debe de estar autenticado en el sistema. | | |
| **Postcondiciones:** | * Los cambios realizados de la compra se actualizarán en el servidor como comprado. | | |
| **Flujo Principal:** | 1. El caso de uso inicia cuando el usuario clic en el botón de “**comprar”.** 2. El sistema nos desplegará un módulo para realizar la compra de las fotografías, mediante un módulo externo de pagos. | | |
| **Flujo(s) alternativo(s):** |  | | |

### **3.3.7. CU07: Modulo de reconocimiento facial.**

#### **3.3.7.1. Diagrama de caso de uso**



#### **3.3.7.2. Descripción del caso de uso**

|  |  |  |  |
| --- | --- | --- | --- |
| **CASO DE USO: Modulo de reconocimiento facial** | | | |
| **ID :** | **CU07.** | | |
| **Creado por:** | Alex Limbert Yalusqui Godoy | **Última modificación por:** | Alex Limbert Yalusqui Godoy |
| **Fecha de creación:** | 22-enero-2017 | **Fecha de última modificación:** | 22-enero-2017 |
| **Actores Primarios:** | Servicio | | |
| **Actores secundarios:** |  | | |
| **Descripción:** | Este es un servicio que se está ejecutando en el servidor verificando nuevas fotografías y realizando análisis mediante reconocimiento facial. | | |
| **Precondiciones:** | * El gestor de base de datos debe de estar iniciado | | |
| **Postcondiciones:** | * Cuando se encuentran coincidencias con el reconocimiento facial. Se generan tablas con los respectivos datos del cliente y evento. | | |
| **Flujo Principal:** | 1. Cuando se inicia el servicio. 2. Se obtiene las fotos no procesadas de los clientes. 3. De cada fotografía se procesa el rostro. 4. Una vez procesada cada fotografía de cliente, se empieza a buscar alguna coincidencia dentro de las fotografías de los eventos 5. Si se encuentra se genera un detalle de ubicación. | | |
| **Flujo(s) alternativo(s):** |  | | |

## **MODELO GENERAL DE CASOS DE USO**



# **CAPITULO 4: ANALISIS**

## **ANALISIS DE LA ARQUITECTURA**

### **IDENTIFICACION DE PAQUETES**

Se identificaron tres paquetes en el sistema.



#### **4.1.1.1. Paquete Gestión de Cliente y ventas.**



#### **4.1.1.2. Paquete de Servicio fotográfico**



#### **4.1.1.3. Paquete de Servicio de reconocimiento facial**



## **ANALISIS DE CASOS DE USO**

### **4.2.1. CU1: Gestionar perfil de usuario.**

### **4.2.2. CU2: Gestionar eventos sociales.**

### **4.2.3. CU3: Gestionar fotografías de perfil.**

### **4.2.4. CU4: Gestionar fotografías de eventos sociales.**

### **4.2.5. CU5: Galería de fotografías de usuarios.**

### **4.2.6. CU6: Gestionar compras de fotografía.**

### **4.2.7. CU7: Modulo de reconocimiento facial.**

# **CAPITULO 5: DISEÑO**

## **5.1. DISEÑO DE LA ARQUITECTURA**

### **5.1.1. Paquete de administración**



### **5.1.2. Paquete de gestión de equipos**



### **5.1.3. Paquete de gestión de personal**

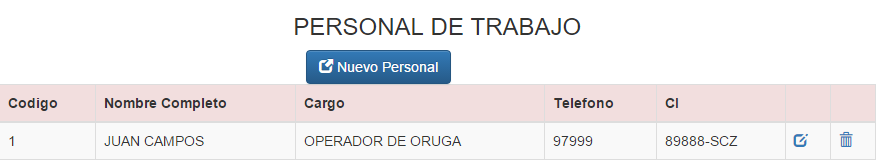


### **5.1.4. Paquete de gestión de producción**

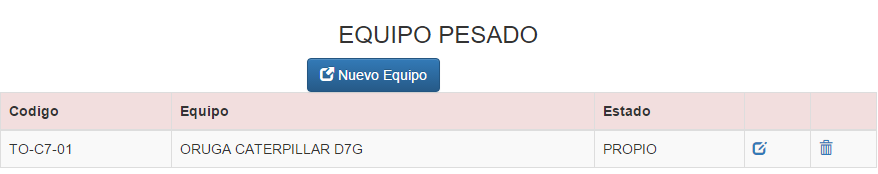


## **5.2. DISEÑO DE LA INTERFAZ**

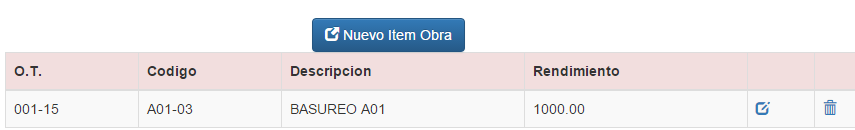
### **5.2.1. CU01: Gestionar personal**

* Vista índice de personal
* Vista del formulario de ingreso de datos

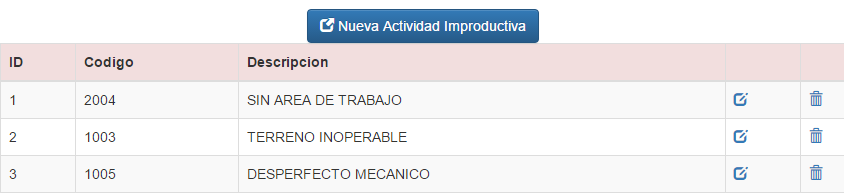
### **5.2.2. CU02: Gestionar equipo pesado**

* Vista índice de equipo pesado
* Vista del formulario de ingreso de datos

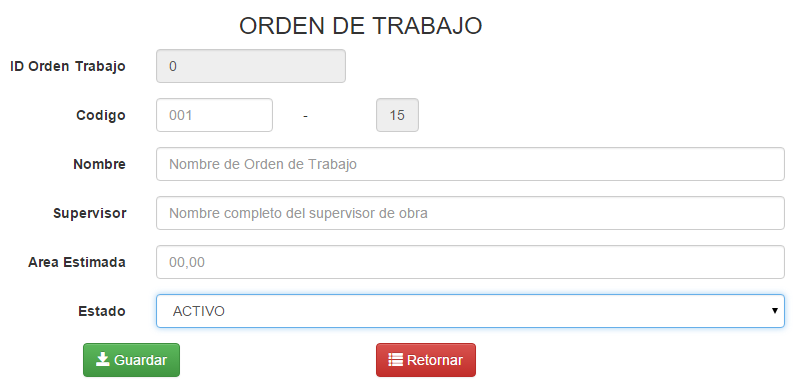
### **5.2.3. CU03: Gestionar ítem obra**

* Vista índice de ítems obras
* Vista del formulario de ingreso de datos

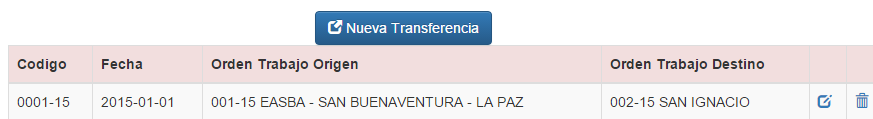
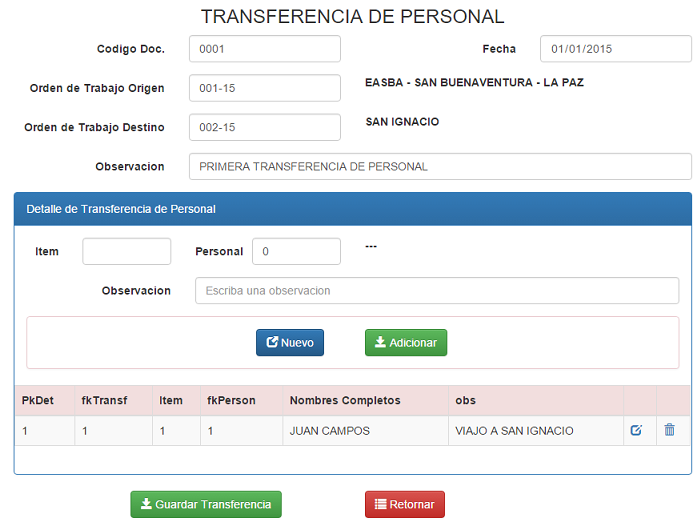
### **5.2.4. CU04: Gestionar actividades improductivas**

* Vista índice de actividades improductivas
* Vista del formulario de ingreso de datos

### **5.2.5. CU05: Gestionar orden de trabajo**

* Vista índice de órdenes de trabajo
* Vista del formulario de ingreso de datos

### **5.2.6. CU06: Gestionar transferencia de personal**

* Vista índice de transferencia de personal
* Vista del formulario de ingreso de datos

### **5.2.7. CU07: Gestionar transferencia de equipo**

* Vista índice de transferencia de equipos
* Vista del formulario de ingreso de datos

### **5.2.8. CU08: Gestionar parte diario de equipo**

* Vista índice de parte diario de equipo
* Vista del formulario de ingreso de datos

### **5.2.9. CU09: Administrar gestiones**

* Vista índice de gestiones
* Vista del formulario de ingreso de datos



### **5.2.10. CU10: Gestionar baja de equipo pesado**

* Vista índice de baja de equipo
* Vista del formulario de ingreso de datos

### **5.2.11. CU11: Gestionar baja de personal**

* Vista índice de baja de personal
* Vista del formulario de ingreso de datos

## **5.3. DISEÑO DE LA BASE DE DATOS**

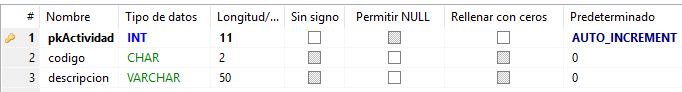
### **5.3.1. Diagrama de clases estáticas**

* Diseño conceptual
* Diseño Logico

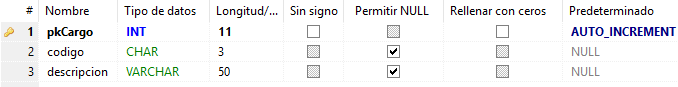
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| spactividad | pk |  |  |  |
|  | pkActividad | codigo | descripcion |  |
|  |  |  |  |  |
| spcargo | pk |  |  |  |
|  | pkCargo | codigo | descripcion |  |
|  |  |  |  |  |
| speqmodelo | pk | fk |  |  |
|  | pkEqModelo | fkEqTipo | codigo | descripcion |
|  |  |  |  |  |
| speqtipo | pk |  |  |  |
|  | pkEqTipo | Código | descripcion |  |

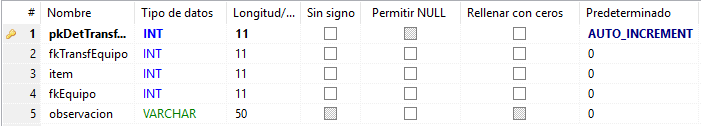
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| spequipo | Pk | fk | fk |  |  |  | fk |  |  |
|  | pkEquipo | fkTipoEquipo | fkModelo | codigo | fkTipoContrato | fechaIngreso | fkOrdenTrabajo | descripcion |  |
|  |  |  |  |  |  |  |  |  |  |
| spgestion | Pk |  |  |  |  |  |  |  |  |
|  | pkGestion | codigo | fechaFin | fechaFin | estado |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| spimproductiva | Pk |  |  |  |  |  |  |  |  |
|  | pkImproductiva | codigo | descripcion |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| spitemobra | Pk | fk | fk | fk |  |  |  |  |  |
|  | pkItemObra | fkOrdenTrabajo | fkPoligono | fkActividad | codigo | descripcion | areaTrab | rendimiento |  |
|  |  |  |  |  |  |  |  |  |  |
| spordentrabajo | Pk |  | fk |  |  |  |  |  |  |
|  | pkOrdenTrabajo | codigo | fkGestion | nombre | supervisor | areaEstimada | estado | data |  |
|  |  |  |  |  |  |  |  |  |  |
| sppersonal | Pk |  |  |  |  |  |  |  | fk |
|  | pkPersonal | fechaIngreso | nombreComp | apellidos | direccion | telefono | ci | fechaNac | fkCargo |
|  |  |  |  |  |  |  |  |  |  |
| sppoligono | Pk | fk |  |  |  |  |  |  |  |
|  | pkPoligono | fkOrdenTrabajo | codigo | descripcion |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| sptransfequipo | Pk |  | fk |  |  |  |  |  |  |
|  | pkTransfEquipo | codigo | fkGestion | fecha | fkOrdenOrigen | fkOrdenDestino | observacion | data | estado |
|  |  |  |  |  |  |  |  |  |  |
| spdettransfequipo | pk | fk |  | fk |  |  |  |  |  |
|  | pkDetTransfEquipo | fkTransfEquipo | item | fkEquipo | observacion |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| sptransfpersonal | pk |  | fk |  |  |  |  |  |  |
|  | pkTransfPersonal | codigo | fkGestion | fecha | fkOrdenOrigen | fkOrdenDestino | observacion | data | estado |
|  |  |  |  |  |  |  |  |  |  |
| dpdettransfpersonal | pk | fk |  |  |  |  |  |  |  |
|  | pkDetTransfPersonal | fkTransfPersonal | item | fkPersonal | observacion |  |  |  |  |

* Diseño fisico

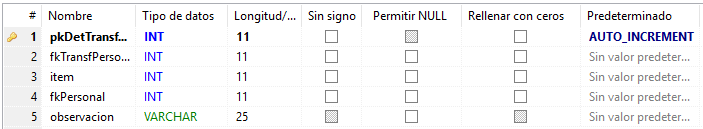
spactividad

spcargo



Spdettransfequipo

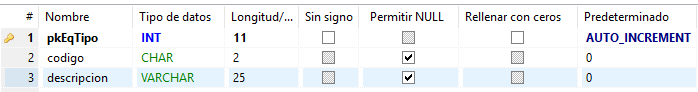
Spdettransfequipo



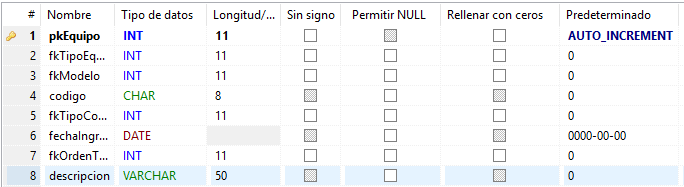
Speqmodelo



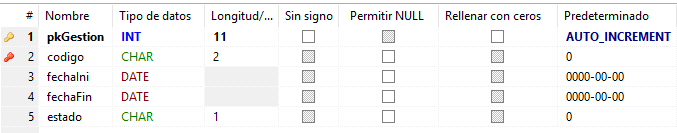
Speqtipo



Spequipo



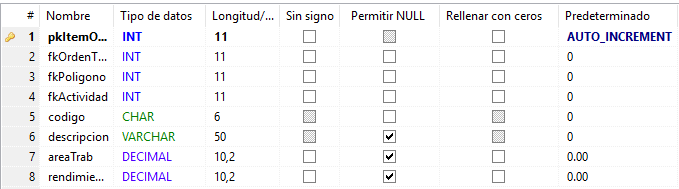
Spgestion



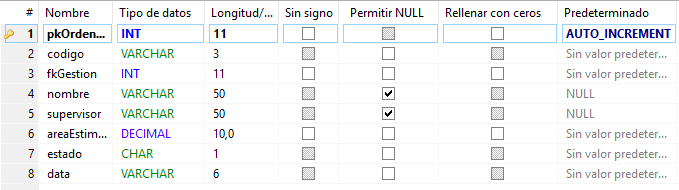
Spimproductiva



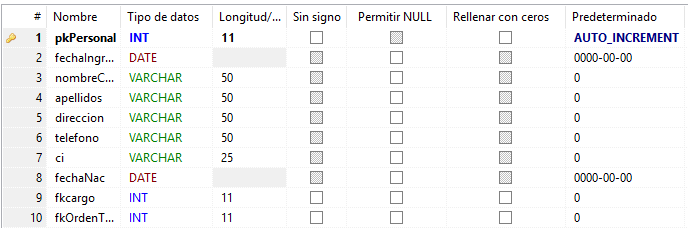
Spitemobra



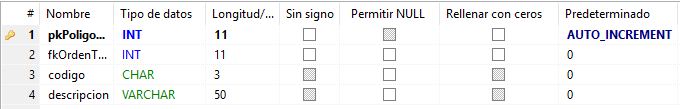
Spordentrabajo



Sppersonal



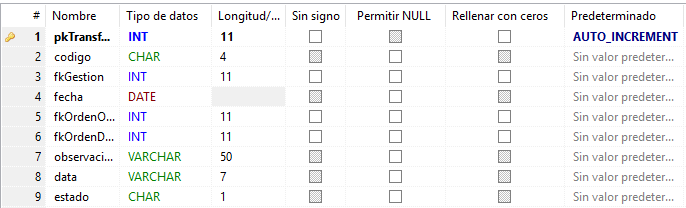
Sppoligono



Sptransfequipo



Sptransfpersonal



**SCRIPT DE BASE DE DATOS**

CREATE TABLE IF NOT EXISTS `spactividad` (

`pkActividad` int(11) NOT NULL AUTO\_INCREMENT,

`codigo` char(2) NOT NULL DEFAULT '0',

`descripcion` varchar(50) NOT NULL DEFAULT '0',

PRIMARY KEY (`pkActividad`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `spcargo` (

`pkCargo` int(11) NOT NULL AUTO\_INCREMENT,

`codigo` char(3) DEFAULT NULL,

`descripcion` varchar(50) DEFAULT NULL,

PRIMARY KEY (`pkCargo`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1

CREATE TABLE IF NOT EXISTS `spdettransfequipo` (

`pkDetTransfEquipo` int(11) NOT NULL AUTO\_INCREMENT,

`fkTransfEquipo` int(11) NOT NULL DEFAULT '0',

`item` int(11) NOT NULL DEFAULT '0',

`fkEquipo` int(11) NOT NULL DEFAULT '0',

`observacion` varchar(50) NOT NULL DEFAULT '0',

PRIMARY KEY (`pkDetTransfEquipo`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `spdettransfpersonal` (

`pkDetTransfPersonal` int(11) NOT NULL AUTO\_INCREMENT,

`fkTransfPersonal` int(11) NOT NULL,

`item` int(11) NOT NULL,

`fkPersonal` int(11) NOT NULL,

`observacion` varchar(25) NOT NULL,

PRIMARY KEY (`pkDetTransfPersonal`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `speqmodelo` (

`pkEqModelo` int(11) NOT NULL AUTO\_INCREMENT,

`fkEqTipo` int(11) DEFAULT '0',

`codigo` char(2) DEFAULT '0',

`descripcion` varchar(25) DEFAULT '0',

PRIMARY KEY (`pkEqModelo`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `speqtipo` (

`pkEqTipo` int(11) NOT NULL AUTO\_INCREMENT,

`codigo` char(2) DEFAULT '0',

`descripcion` varchar(25) DEFAULT '0',

PRIMARY KEY (`pkEqTipo`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `spequipo` (

`pkEquipo` int(11) NOT NULL AUTO\_INCREMENT,

`fkTipoEquipo` int(11) NOT NULL DEFAULT '0',

`fkModelo` int(11) NOT NULL DEFAULT '0',

`codigo` char(8) NOT NULL DEFAULT '0',

`fkTipoContrato` int(11) NOT NULL DEFAULT '0',

`fechaIngreso` date NOT NULL DEFAULT '0000-00-00',

`fkOrdenTrabajo` int(11) NOT NULL DEFAULT '0',

`descripcion` varchar(50) NOT NULL DEFAULT '0',

PRIMARY KEY (`pkEquipo`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `spgestion` (

`pkGestion` int(11) NOT NULL AUTO\_INCREMENT,

`codigo` char(2) NOT NULL DEFAULT '0',

`fechaIni` date NOT NULL DEFAULT '0000-00-00',

`fechaFin` date NOT NULL DEFAULT '0000-00-00',

`estado` char(1) NOT NULL DEFAULT '0',

PRIMARY KEY (`pkGestion`),

UNIQUE KEY `codigo` (`codigo`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `spimproductiva` (

`pkImproductiva` int(11) NOT NULL AUTO\_INCREMENT,

`codigo` varchar(4) DEFAULT NOT NULL,

`descripcion` varchar(50) DEFAULT NOT NULL,

PRIMARY KEY (`pkImproductiva`),

UNIQUE KEY `codigo` (`codigo`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1

CREATE TABLE IF NOT EXISTS `spitemobra` (

`pkItemObra` int(11) NOT NULL AUTO\_INCREMENT,

`fkOrdenTrabajo` int(11) NOT NULL DEFAULT '0',

`fkPoligono` int(11) NOT NULL DEFAULT '0',

`fkActividad` int(11) NOT NULL DEFAULT '0',

`codigo` char(6) NOT NULL DEFAULT '0',

`descripcion` varchar(50) DEFAULT '0',

`areaTrab` decimal(10,2) DEFAULT '0.00',

`rendimiento` decimal(10,2) DEFAULT '0.00',

PRIMARY KEY (`pkItemObra`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `spordentrabajo` (

`pkOrdenTrabajo` int(11) NOT NULL AUTO\_INCREMENT,

`codigo` varchar(3) NOT NULL,

`fkGestion` int(11) NOT NULL,

`nombre` varchar(50) DEFAULT NULL,

`supervisor` varchar(50) DEFAULT NULL,

`areaEstimada` decimal(10,0) NOT NULL,

`estado` char(1) NOT NULL,

`data` varchar(6) NOT NULL,

PRIMARY KEY (`pkOrdenTrabajo`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `sppersonal` (

`pkPersonal` int(11) NOT NULL AUTO\_INCREMENT,

`fechaIngreso` date NOT NULL DEFAULT '0000-00-00',

`nombreComp` varchar(50) NOT NULL DEFAULT '0',

`apellidos` varchar(50) NOT NULL DEFAULT '0',

`direccion` varchar(50) NOT NULL DEFAULT '0',

`telefono` varchar(50) NOT NULL DEFAULT '0',

`ci` varchar(25) NOT NULL DEFAULT '0',

`fechaNac` date NOT NULL DEFAULT '0000-00-00',

`fkcargo` int(11) NOT NULL DEFAULT '0',

`fkOrdenTrabajo` int(11) NOT NULL DEFAULT '0',

PRIMARY KEY (`pkPersonal`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `sppoligono` (

`pkPoligono` int(11) NOT NULL AUTO\_INCREMENT,

`fkOrdenTrabajo` int(11) NOT NULL DEFAULT '0',

`codigo` char(3) NOT NULL DEFAULT '0',

`descripcion` varchar(50) NOT NULL DEFAULT '0',

PRIMARY KEY (`pkPoligono`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `sptransfequipo` (

`pkTransfEquipo` int(11) NOT NULL AUTO\_INCREMENT,

`codigo` char(4) NOT NULL DEFAULT '0',

`fkGestion` int(11) NOT NULL DEFAULT '0',

`fecha` date NOT NULL DEFAULT '0000-00-00',

`fkOrdenOrigen` int(11) NOT NULL DEFAULT '0',

`fkOrdenDestino` int(11) NOT NULL DEFAULT '0',

`observacion` varchar(50) NOT NULL DEFAULT '0',

`data` varchar(7) NOT NULL DEFAULT '0',

`estado` char(1) NOT NULL DEFAULT '0',

PRIMARY KEY (`pkTransfEquipo`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `sptransfpersonal` (

`pkTransfPersonal` int(11) NOT NULL AUTO\_INCREMENT,

`codigo` char(4) NOT NULL,

`fkGestion` int(11) NOT NULL,

`fecha` date NOT NULL,

`fkOrdenOrigen` int(11) NOT NULL,

`fkOrdenDestino` int(11) NOT NULL,

`observacion` varchar(50) NOT NULL,

`data` varchar(7) NOT NULL,

`estado` char(1) NOT NULL,

PRIMARY KEY (`pkTransfPersonal`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

## **5.4. DISEÑOS PROCIDEMENTALES**

### **5.4.1. CU01: Gestionar personal**

* Diagrama de clases dinámicas
* Diagrama de secuencia

### **5.4.2. CU02: Gestionar equipo pesado**

* Diagrama de clases dinámicas
* Diagrama de secuencia

### **5.4.3. CU03: Gestionar ítem obra**

* Diagrama de clases dinámicas
* Diagrama de secuencia

### **5.4.4. CU04: Gestionar actividades improductivas**

* Diagrama de clases dinámicas
* Diagrama de secuencia

### **5.4.5. CU05: Gestionar orden de trabajo**

* Diagrama de clases dinámicas
* Diagrama de secuencia

### **5.4.6. CU06: Gestionar transferencia de personal**

* Diagrama de clases dinámicas
* Diagrama de secuencia

### **5.4.7. CU07: Gestionar transferencia de equipo**

* Diagrama de clases dinámicas
* Diagrama de secuencia

### **5.4.8. CU08: Gestionar parte diario de equipo**

* Diagrama de clases dinámicas
* Diagrama de secuencia

### **5.4.9. CU09: Administrar gestiones**

* Diagrama de clases dinámicas
* Diagrama de secuencia

### **5.4.10. CU10: Gestionar baja de equipo pesado**

* Diagrama de clases dinámicas
* Diagrama de secuencia

### **5.4.11. CU11: Gestionar baja de personal**

* Diagrama de clases dinámicas
* Diagrama de secuencia

CU01: Gestionar Personal. Ok 17:30

CU02: Gestionar Equipo pesado. Ok 18:00

CU03: Gestionar ítem obra (actividades productivas). Ok 18:30

CU04: Gestionar actividades improductivas. Ok 19:00

CU05: Gestionar Orden de trabajo. Ok 19:30

CU06: Gestionar transferencia de personal. Ok+ 20:00

CU07: Gestionar transferencia de equipo pesado. Ok+ 20:30

CU08: Gestionar parte diario de equipo. no

CU09: Administrar gestiones de periodo. Ok 21:00

CU10: Gestionar baje de equipo pesado. no

CU11: Gestionar baja de personal. no