# http://blogs.espe.edu.ec/wp-content/uploads/2013/09/LOGO-PRINCIPAL7.png

**Nombres: Espinoza Borys Jair**



**Nivel: 7º**

**Carrera: Software**

**Materia: Proyecto Integrador III**

**Tema**:  **Documento de Arquitectura Software**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revisión** | **Fecha** | **Modificación Realizada** | **Realizado Por:** |
| 0 | 15 de junio del  2018 | Versión Inicial. | Borys Jair Espinoza |
|  |  |  |  |
|  |  |  |  |

**TABLA DE CONTENIDO**

**GLOSARIO DE TÉRMINOS .............................................................................................................. 1-4**

1. **INTRODUCCIÓN ......................................................................................................................... 1-5**
   1. OBJETO DEL DOCUMENTO. ......................................................................................................... 1-5
   2. EQUIPO DE TRABAJO. ................................................................................................................. 1-5
   3. MATERIAL REVISADO. ................................................................................................................. 1-6
2. **DESCRIPCIÓN GENERAL DEL SISTEMA............................................................................. 2-6**
   1. Proceso..................................................................................................................................... 2-6
3. **ARQUITECTURA…………………………............................................................................... 3-6**
   1. Definiciones de Arquitectura ................................................................................................... 3-6

*3.2.1* *Capas de la Arquitectura: ............................................................................................. 3-7*

* 1. Arquitectura de Infraestructura

*3.2.1* *Plataforma de software de base.......................................................................................... 3-7*

* 1. Arquitectura Lógica

3.4. Capas

3.5. Arquitectura Física

3.6. Herramientas de desarrollo

1. **MEJORES PRÁCTICAS ........................................................................................................... 3-10**
   1. GIT ......................................................................................................................................... 3-11
   2. Trello ......................................................................................................................................... 3-11
2. **CALIDAD .................................................................................................................................... 5-12**
3. **DOCUMENTACIÓN .................................................................................................................. 6-12**
4. **ANEXOS ...................................................................................................................................... 7-13**

# **GLOSARIO DE TÉRMINOS**

**Definiciones:**

Dentro del contexto del presente documento se definen los siguientes términos:

**Paquetes de datos:** Agrupa todos los datos de un ciudadano en una institución que pueden ser consultados a través del servicio web de Interoperabilidad.

**JEE 7:** Es la versión más reciente y estable de un conjunto de especificaciones, funcionalidades y estándares orientadas al desarrollo de aplicaciones empresariales, escalables, portables, y móviles utilizando el lenguaje de programación Java.

**PrimeFaces:** es un framework que utiliza la tecnología JavaServer Faces (JSF) para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE.

**EJB:** Enterprise JavaBeans proporcionan un modelo de componentes distribuidos estándar del lado del servidor que le permite al programador abstraerse de los problemas generales de una aplicación empresarial para centrarse en el desarrollo de la lógica de negocio en sí.

**CDI:** Contexts and Dependency Injection for Java EE, es un conjunto de servicios contextuales que facilitan a los desarrolladores utilizar Enterprise JavaBeans junto a los JavaServer Faces en aplicaciones web.

**JPA:** Es una herramienta del estándar de Java que realiza el Mapeo objeto-relacional (ORM) entre una base de datos relacional tradicional y el modelo de objetos de una aplicación.

**AJAX:** Es una tecnología asíncrona, en que los datos se solicitan mediante Java al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página maquetando las peticiones y respuestas en formato XML.

**XML**: Es un lenguaje de maquetado de datos. Permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

**DDD:** Domain-Driven Design es un enfoque para el desarrollo de software, no es una tecnología ni una metodología. Provee una estructura de prácticas y terminologías para tomar decisiones de diseño que enfoquen y aceleren la creación y mantenimiento de dominios complejos en los proyectos de software.

## 1. INTRODUCCIÓN

En nuestro sitio web se ha optado por un modelo de programación por capas. Las capas que contiene nuestro diseño son la capa de presentación, la de negocio y la de datos. Este modelo propone la separación de la lógica de negocio de la de diseño o la de datos.

Optar por este modelo nos permite tener claramente separados las tres funcionalidades diferentes. Esto nos permite ahorrarnos inconvenientes en cuanto a escalabilidad, disponibilidad y seguridad del sistema, tanto como para futuras mejoras o ampliaciones, permitiendo un avance mucho más rápido en el conocimiento de la aplicación.

* 1. **Objeto del Documento.**

Este documento es actualizable de acuerdo a los cambios técnicos de arquitectura que vayan apareciendo. Se detalla el equipo que mantiene la infraestructura técnica y los requerimientos de cambios que el equipo pudiera tener. El objetivo del documento es mantener organizada la Arquitectura Técnica Organizacional y proveer una fuente de referencia para los analistas y diseñadores de la aplicación.

Los analistas y desarrolladores de la aplicación utilizarán este documento para propósitos informativos de entendimiento de la actual arquitectura, y para los propósitos del análisis para discernir si la arquitectura, como es, apoyará sus requisitos de la aplicación. Si se exigen cambios a la arquitectura, estos cambios deben revisarse con el equipo de Infraestructura para su resolución. El diseñador de la aplicación hará las modificaciones apropiadas entonces a este documento.

* 1. **Equipo de Trabajo.**

Las personas que directa o indirectamente intervinieron fueron las siguientes:

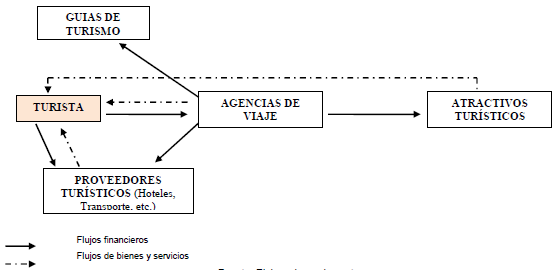
|  |  |
| --- | --- |
| **Nombre** | **Cargo** |
| Espinoza Zamora Borys | Analista de Infraestructura |
| Espinoza Zamora Borys | Directora de Gestión y Registro (Administradora del Contrato) |
| Espinoza Zamora Borys | Coordinador de Infraestructura y Seguridad Informática |
| Espinoza Zamora Borys | Asesora de Coordinación de Infraestructura y Seguridad Informática |
| Espinoza Zamora Borys | Directora de Tecnología y Desarrollo |

**1.3 Material Revisado.**

* Estándares y herramientas para el desarrollo de aplicaciones Java.

## 2. DESCRIPCIÓN GENERAL DEL SISTEMA

### 2.1. Proceso



## 3. ARQUITECTURA

### 3.1. Definiciones de Arquitectura

La arquitectura seleccionada para el desarrollo del sistema es una arquitectura n-capa con cliente WEB.

Identificamos como Usuarios comunes aquellos que acceden al sistema para hacer uso de la funcionalidad de la aplicación, e identificamos como usuarios administradores del sistema a aquellos que acceden para realizar mantenimiento de la aplicación, administrar seguridades, catálogos, entre otros.

Ambos tipos de usuarios dispondrán de una aplicación cliente web, que se comunicará con una aplicación de servidor la cual implementará las reglas del negocio y a su vez será la de intermediaria entre la aplicación cliente web y la base de datos.

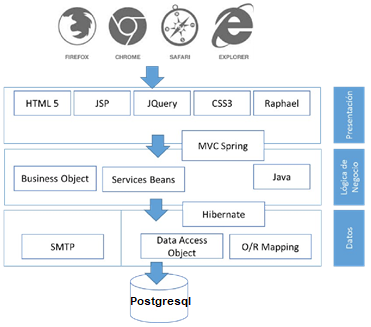
La interfaz de usuario será un “Client Rich” ya que se requiere en interfaz de usuario interactiva con el usuario, experiencias de usuario óptimo para el manejo de gráficos y alto rendimiento.

### 3.1.1. Capas de la Arquitectura

El sistema CAD-CAM ha sido visionado como una aplicación web bajo la especificación Java Servlet, con una arquitectura de N-capas. Estará compuesto por las siguientes capas:

* Presentación: bajo patrón de diseño Modelo-Vista-Controlador:
* Vistas: Páginas web JSP con tag libraries de SpringMVC que despliegan formularios al usuario.
* Controladores: Controladores de SpringMVC, que reciben y procesan las solicitudes del usuario.
* Modelos: objetos reutilizados de la capa de modelo de dominio.
* Lógica de Negocios:
* Capa de servicios: componentes encargados de ejecutar procesos complejos de la lógica de negocio, que interactuarán con los objetos del modelo de dominio.
* Capa de modelo de dominio: componentes con la estructura conceptual que representa el dominio de la aplicación, en la forma de JavaBeans tradicionales.
* Datos:
* Capa de implementación de acceso a datos: ORM Hibernate.

### 3.2. Arquitectura de Infraestructura



### 3.2.1. Plataforma de software de base

Cliente:

Sistema Operativo

Multiplataforma

Browser

Internet Explorer, Chrome, Firefox

Tecnologías

utilizadas

HTML5

CSS3

jQuery

Raphael

Lenguajes Utilizados

JavaScript

Sistema Operativo

Servidor de aplicación:

**Sistema Operativo**

Multiplataforma

**Application Server**

Tomcat

**Lenguajes Utilizados**

Java J2EE

**Framworks**

**Utilizados**

Spring MVC, Hibernate

Servidor de base de datos:

Multiplataforma

Base de Datos

Postgresql

### 3.3. Arquitectura Lógica

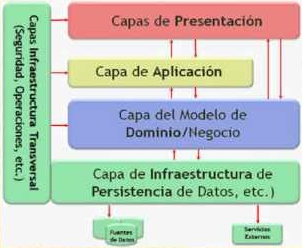
La arquitectura de tres capas por la que hemos optado en nuestro sitio web separa la implementación y diseño de la aplicación en 3 capas o niveles. Por una parte, tendremos la capa donde se gestiona la información del sistema, es decir la base de datos. Seguidamente tendremos la lógica de negocio y por último, tenemos la última capa, la interfaz gráfica o capa de presentación.

A continuación, un breve resumen de lo que contiene cada una de las capas.

3.3.1. Capa de datos: es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio

3.3.2. Capa de negocio: es donde residen las funciones que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.

3.3.3. Capa de presentación: la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario. Esta capa se comunica únicamente con la capa de negocio.



Como ventajas de esta arquitectura destacaremos las siguientes:

1. Las llamadas de la interfaz de usuario en la estación de trabajo al servidor de capa intermedia son más flexibles que el diseño de dos capas, ya que la estación solo necesita transferir parámetros a la capa intermedia.
2. Con la arquitectura de tres capas la interfaz del cliente no es requerida para comprender o comunicarse con el receptor de los datos, por lo tanto, esa estructura de datos puede ser modificada sin cambiar la interfaz de usuario.
3. El código de la capa de negocio puede ser reutilizado por otras funciones si está diseñado en formato modular.
4. La separación de roles en tres capas hace más fácil reemplazar o modificar una capa sin afectar a los módulos restantes.

**3.4. Capas**

**3.4.1.** **Capa de presentación**

Como ya hemos comentado en el punto anterior esta capa se encarga de proveer una interfaz entre el sistema y el usuario. Esta capa es la responsable de la comunicación de información al usuario por parte del sistema y viceversa, manteniendo ésta a su vez una comunicación exclusiva con la capa de negocio.

Este apartado va a describir el diseño elegido para la capa de presentación del sitio web del recomendado. El sitio web dispone de una interfaz sencilla y funcional para que el usuario pueda actuar con el sistema de una forma fácil y amigable.

**3.4.2. Capa de negocio**

Esta capa contiene los procesos y funciones a realizar. La capa de negocio se encuentra en medio de las otras dos capas y esto hace que comparta la información con estas. Esto significa que debe recoger la información introducida por el usuario en la capa de presentación, realizar las funciones necesarias, preguntarle a la capa de datos por esa información y devolverá en forma que la capa de presentación la entienda y la pueda mostrar.

La capa de negocios en nuestro sitio web contiene las siguientes funciones:

• Administra las variables de sesión de todo el sitio web. Las inicializa, las guarda, las muestra o hace comprobaciones con la información recibida.

• Recoge los parámetros de las consultas enviadas desde la capa de presentación por método GET o POST.

• Realiza la conexión a la base de datos.

• Ejecutar sentencias SQL como consultas, inserciones o borrado de datos directamente en la base de datos.

• Empaqueta la información recibida de la base de datos en arrays.

• Hace la gestión de errores del sitio.

• Programación de algoritmos para diferentes funciones.

• Comprobación y verificación de usuarios.

• Registro de nuevos usuarios en la base de datos.

• Eliminar las variables de sesión al terminar la sesión de usuario.

• Insertar nuevos registros en algunas tablas.

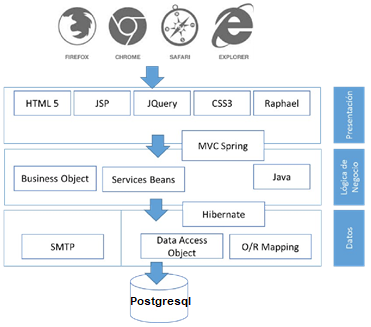
**3.4.3 Capa de datos**

Por último, en la capa de datos es donde se almacenan todos los datos. Desde aquí salen los datos que llegarán, a través de la capa de negocio, hasta el usuario a través de la capa de presentación.

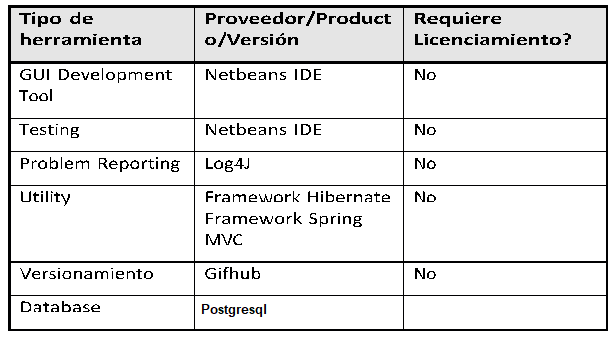
Nosotros utilizaremos un modelo de base de datos SQL y podremos tanto acceder a él mediante el gestor de la base de datos como desde las propias funciones que nos ofrece el sitio web.

El modelo que mostramos a continuación describe de una manera gráfica la base de datos que se encuentra detrás de nuestro sitio web. En él se detalla la composición de la base de datos en sus tablas como su estructura.

### 3.5 Arquitectura Física



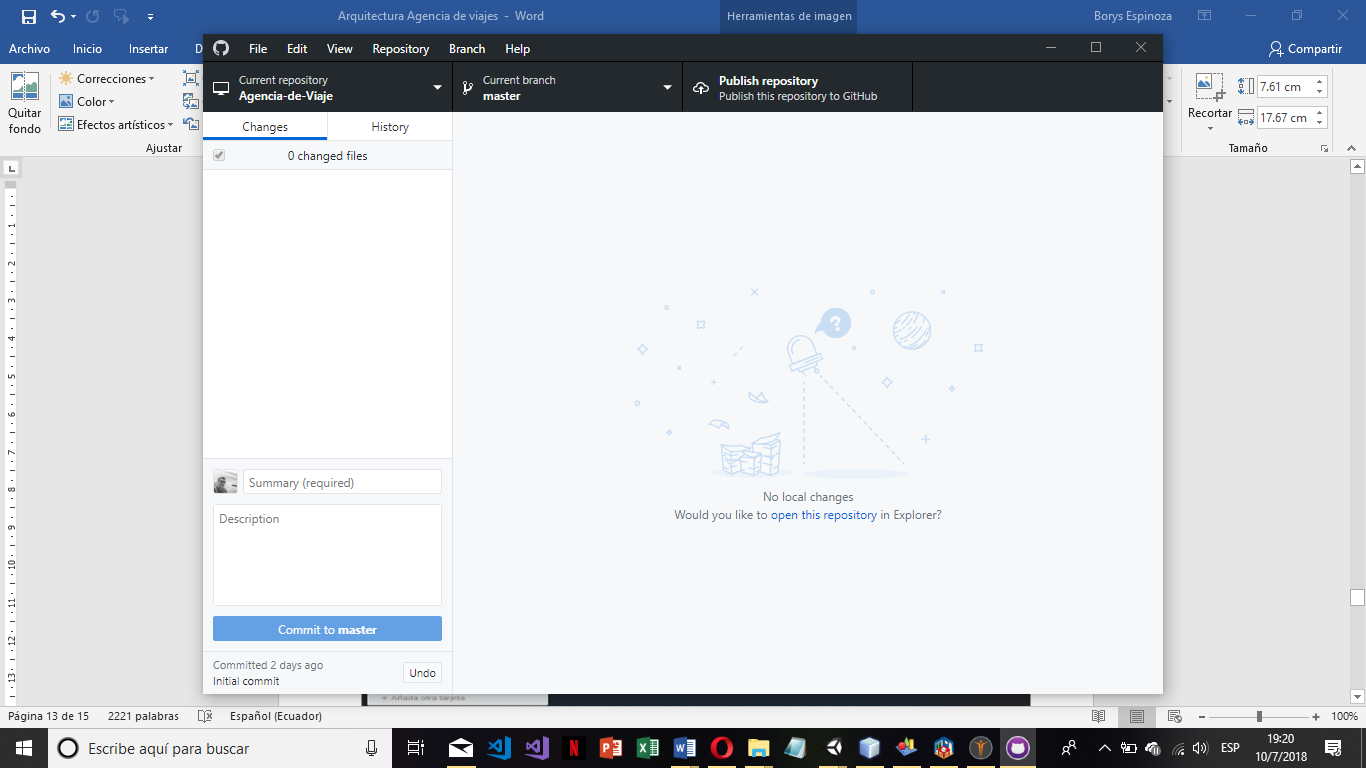
### 3.6 HERRAMIENTAS DE DESARROLLO



## 4. MEJORES PRÁCTICAS

### 4.1. Git

Herramienta de versionamiento de código que permite llevar el control y auditoría de los cambios que realiza cada desarrollador, y cambiar fácilmente de versión para contar siempre con código estable listo para su uso en producción.

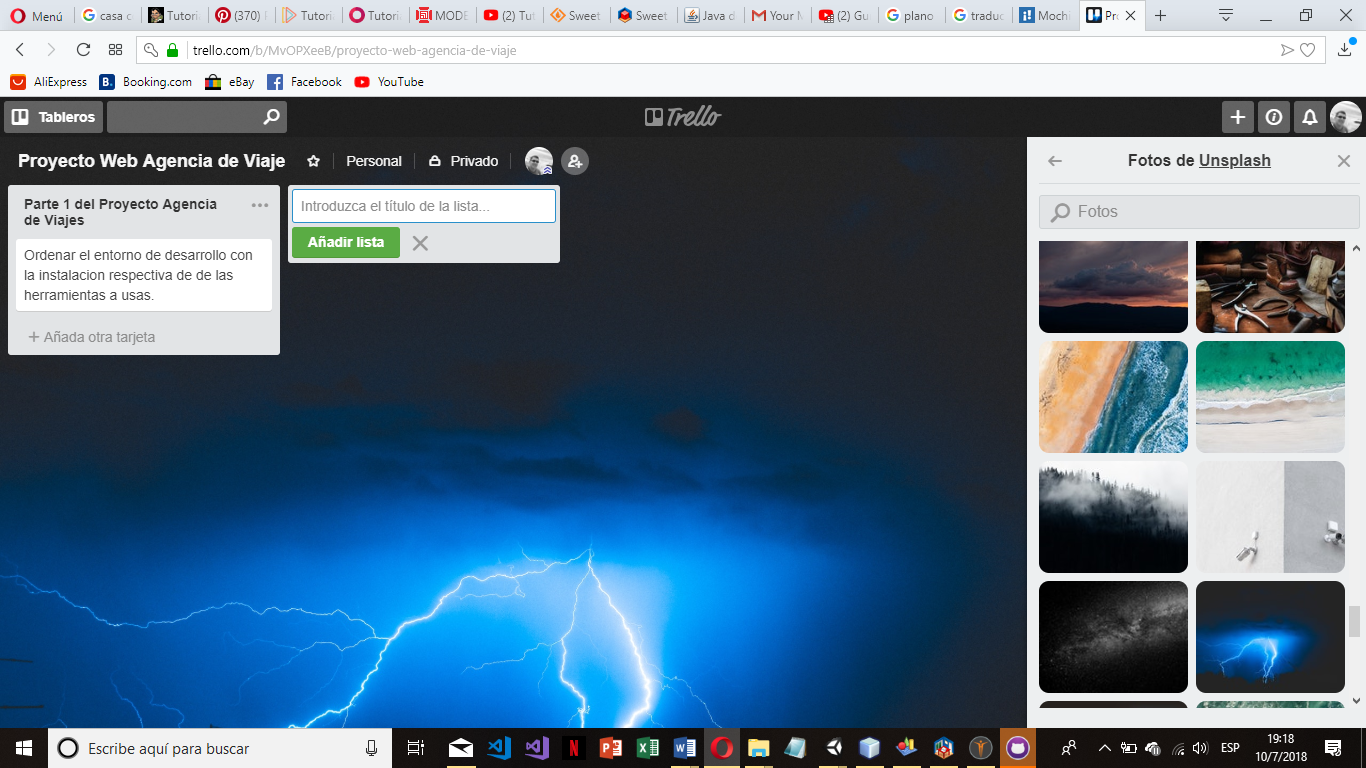


### 4.2. Trello

Trello es un software de administración de proyectos con interfaz web, cliente para iOS y Android para organizar proyectos.

Empleando el sistema kanban, para el registro de actividades con tarjetas virtuales organiza tareas, permite agregar listas, adjuntar archivos, etiquetar eventos, agregar comentarios y compartir tableros.

Trello es un tablón virtual en el que se pueden colgar ideas, tareas, imágenes o enlaces. Es versátil y fácil de usar pudiendo usarse para cualquier tipo de tarea que requiera organizar información.



## 5. CALIDAD

Las consideraciones arquitectónicas y de diseño mostradas en este documento contribuyen a elevar la calidad del servicio de la siguiente manera:

***Escalabilidad y Confiabilidad:*** apoyada por el diseño modular y la separación en componentes, esta permite distribuir los componentes entre varios servidores en caso de ser requerido.

***Mantenimiento:*** Los patrones arquitectónicos que se usan permiten que la solución sea lo suficientemente flexible separando claramente cada parte funcional de la solución para soportar cambios en una manera consistente y sencilla.

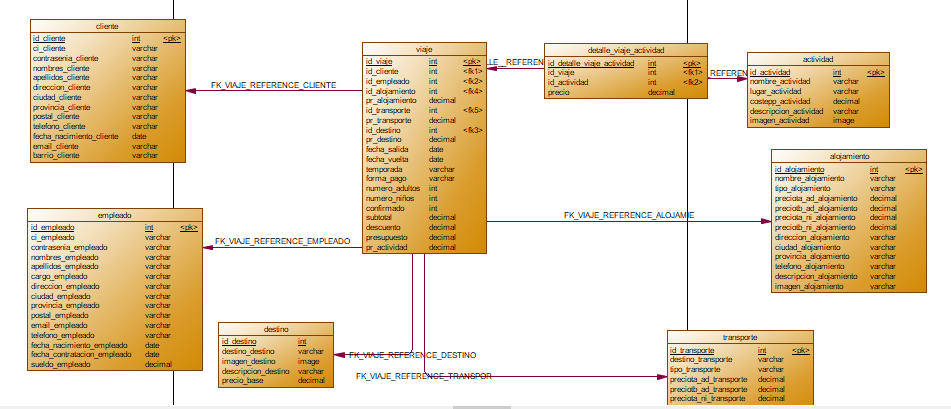
## 6. DOCUMENTACIÓN

Se listan los documentos a entregar junto con el proyecto:

* Diagrama de Arquitectura.
* Diagrama Físico de Base de Datos.
* Manual Técnico de la Herramienta.
* Manual de Usuario.
* Plan de Pruebas.
* Catálogo de Errores

## 7. Anexos

### 7.1 Modelo Físico



### 7.2 Diagrama de Clases

