

Arquitectura de Software

Versión <1.0>

Chávez Salas, Joao Franco Emanuel
Cutipa Ayala, Shyndell Ayelén
Hilares Angelo, Maryori Lizeth
Huayllasco Carlos, Edward Luis
Huertas Canaza, Jim Leonardo
Pumaraime Espinoza, Dennis
Romero Chacón, Solange Aracely
Sante Taipe, Luis Felipe
Villanueva Guerrero, Luisa
Macedo Huaman, Vanessa Mayra

Arequipa - Perú

2021

ARQUITECTURA DE SOFTWARE

1. Introducción
- 1.1 Finalidad
- 1.2 Alcance
- 1.3 Definiciones, Acrónimos y Abreviaturas
- 1.4 Visión General
2. Representación Arquitectural
3. Metas y Restricciones de la Arquitectura
4. Visión de Casos de Uso
5. Visión Lógica
- 5.1 Visión General
6. Visión de implantación
7. Visión de implementación
8. Visión de datos

1. Introducción

Este documento es una Especificación de Requisitos Software (ERS) para el Sistema de Publicación de Proceedings de Eventos. Esta especificación se ha estructurado basándose en las directrices dadas por el estándar IEEE Práctica Recomendada para Especificaciones de Requisitos Software ANSI/IEEE 830, 1998.

1.1. Propósito

El presente documento tiene como propósito definir las especificaciones funcionales, no funcionales para el desarrollo de un sistema de información web que permitirá gestionar distintos procesos de eventos, en específico de eventos de la carrera de Ciencia de la Computación. Éste será utilizado por estudiantes, profesores y público en general.

1.2. Alcance

Esta especificación de requisitos está dirigida al usuario del sistema, para continuar con el desarrollo de aplicaciones divulgativas sobre eventos de la escuela y para profundizar en la automatización de ésta, la cual tiene por objetivo principal el gestionar los distintos eventos académicos para su correcta divulgación y futura participación en ellos por parte del público.

1.3. Definiciones, acrónimos y abreviaturas

Nombre	Descripción
Usuario	Persona que usará el sistema para gestionar procesos
Sistema	Recursos que permitirán desarrollar las funcionalidades del software.
Administrador	Persona con permisos dentro del sistema que le permiten controlar y supervisar el desenvolvimiento del software
Servidores	Conjunto de computadoras capaces de responder las peticiones de los usuarios y administradores en el sistema.
Interfaz	Es la forma de comunicación entre el sistema y los usuarios, o entre sistemas, de forma que les permita interactuar.
Lenguaje	Lenguaje creado para transmitir información a través del sistema. Incluye también el lenguaje de programación empleado.
Módulo	Es una porción del software que cumplirá una tarea definida.
Rendimiento	Cantidad de trabajo que soporta el software sin alterar su funcionamiento.
Fiabilidad	Probabilidad de que el sistema funcione correctamente.
Disponibilidad	Capacidad del sistema para ser accesible y utilizable cuando los usuarios lo requieran.
Seguridad	Capacidad del sistema de proteger la información de los usuarios y la integridad del propio sistema.
Mantenibilidad	Característica del sistema que le permite ser modificado o reparado efectivamente.
Portabilidad	Propiedad por la que el software puede ser utilizado por diferentes tipos de equipos.

SIS-I	Sistema de Información Web para la Gestión de Procesos Administrativos y Académicos
ERS	Especificación de Requisitos Software
RF	Requerimiento Funcional
RNF	Requerimiento No Funcional
FTP	Protocolo de Transferencia de Archivos
TCP/IP	Acrónimo de Transmisión Control Protocol/Internet Protocol (protocolo de control de transmisiones/protocolo de Internet), protocolos usados para el control de la transmisión en Internet. Permite que diferentes tipos de ordenadores o computadoras se comuniquen a través de redes heterogéneas.

1.4 Visión General

En este documento se presentan diferentes vistas arquitectónicas de cómo debe comportarse el Sistema de Publicación de Proceedings de Eventos en diferentes procesos, cómo debe implantarse e implementar, y las restricciones de desempeño y calidad.

2. Representación Arquitectural

- **Visión de casos de uso**

Presenta las características arquitectónicas importantes y los usuarios del sistema.

- **Visión lógica**

Describe las clases y su organización y presenta el patrón arquitectónico que debe usarse para el desarrollo del sistema.

- **Visión de procesos**

Muestra el patrón de comportamiento del sistema ante diferentes acciones del usuario.

- **Visión de implantación**

Describe la estructura del entorno donde se instalará el software.

- **Visión de implementación**

Ilustra la distribución del procesamiento a través de un conjunto de nodos en el sistema, incluida la distribución física de procesos y subprocesos.

3. Metas y restricciones de la Arquitectura

- Interfaz para ser usada con internet.
- Interfaces de uso intuitivas para poder navegar por el aplicativo.
- Lenguajes y tecnologías en uso: HTML, JAVA, CSS, JavaScript.
- Los servidores deben ser capaces de atender consultas que nos permita filtrar resultados, por ejemplo la búsqueda de eventos o del mismo modo poder visualizar trabajos.

- El aplicativo debe contar con un sistema de validación de inicio de sesión.
- El sistema se diseñará según un modelo MVC.
- El sistema deberá tener un diseño e implementación sencilla y respete todas, independiente de la plataforma o del lenguaje de programación.

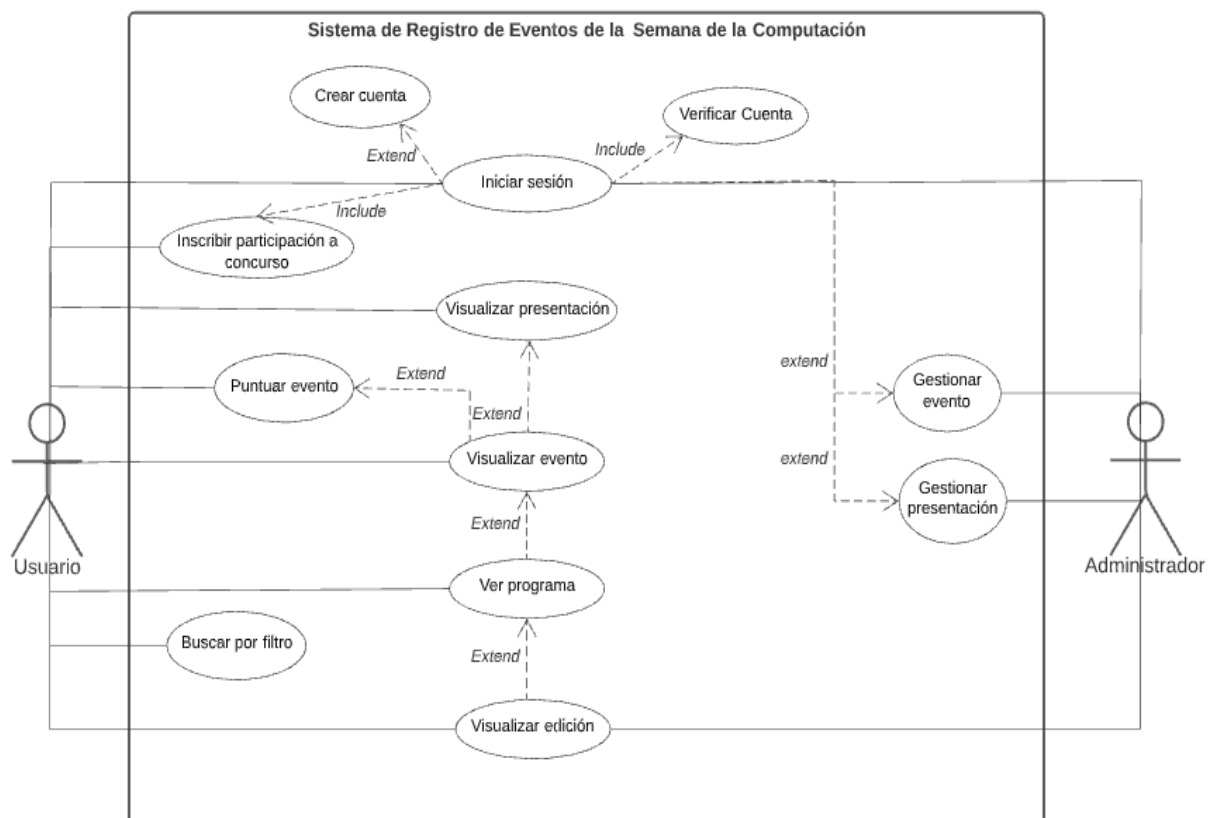
4. Visión de casos de uso

El propósito de la vista del caso de uso es dar un contexto adicional en torno al uso del sistema y las interacciones entre sus componentes. A efectos de este documento, cada componente se considera un actor del caso de uso. En el apartado 4.1 se enumeran los actores actuales y se ofrece una breve descripción de cada uno de ellos en el contexto general de uso del sistema. En la sección 4.2, se describen e ilustran los casos de uso más comunes utilizando diagramas de casos de uso UML para aclarar las interacciones entre los componentes.

4.1. Actores

- **Usuario:** El usuario puede ser un profesor, estudiante o visitante que quiera visualizar información del sistema (visualizar videos, proyectos y eventos).
- **Administrador:** El administrador se encarga del control y manejo del sistema en general.

4.2. Diagrama de Casos de Uso



Título	Inicio de sesión
Objetivo	El usuario desea registrar asistencia a un evento.
Descripción	Se requiere haberse registrado previamente en el sistema (tener una cuenta) y que el sistema compruebe que la contraseña coincide con la cuenta asociada. El usuario se inscribe en el concurso que desea participar. El administrador puede gestionar los eventos.
Actor	Usuario, Administrador

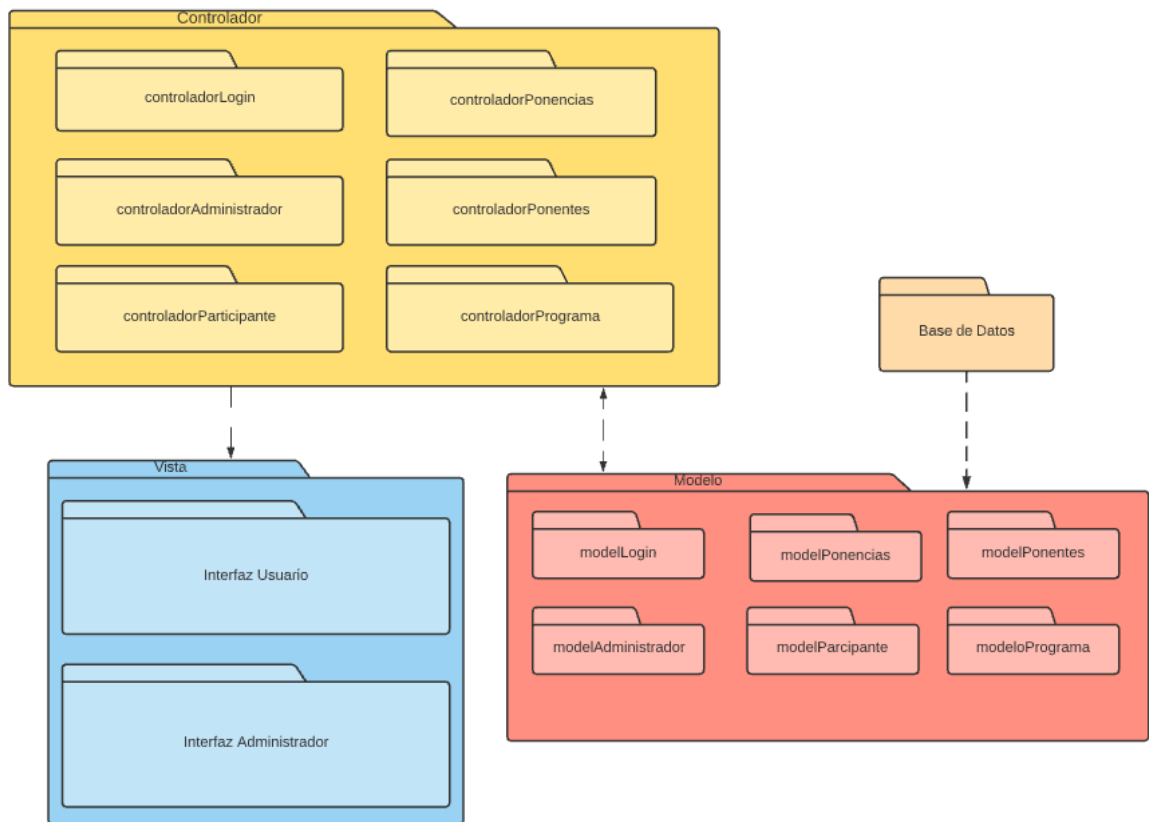
Título	Crear Cuenta
Objetivo	El usuario desea registrarse en el sistema/crear una cuenta
Descripción	Se requiere que el correo que el usuario ingrese no haya sido registrado previamente.El usuario puede inscribirse en el concurso al que desea participar.
Actor	Usuario

Título	Gestionar eventos
Objetivo	El administrador desea publicar, editar, añadir o eliminar un determinado evento.
Descripción	Se requiere que el administrador haya ingresado al sistema, luego podrá actualizar las ponencias, insertar nuevas ponencias o eliminarlas si desea .
Actor	Administrador

Título	Visualizar ediciones y cronograma de eventos
Objetivo	El usuario desea visualizar las ediciones de la semana de la Computación, además desea ver la programación de la edición elegida y los eventos que contiene.
Pre-condición	Se requiere que el usuario ingrese a la página principal, luego se muestra la programación y eventos de una determinada edición elegida por el usuario.
Actor	Usuario, Administrador

5. Visión Lógica

5.1. Visión General



Elementos: Los elementos son los siguientes

Paquete de vista:

El paquete de vista proporciona al usuario la interfaz gráfica para que interactúe con el sistema. Este será implementado mediante archivos HTML, CSS, **JavaScript** y otros compatibles con el navegador. El objetivo principal de esta capa es que el usuario brinde la información necesaria al sistema para realizar las tareas básicas, más adelante detalladas.

Paquete de modelo:

Es donde se trabaja con los datos, por tanto contendrá mecanismos para acceder a la información y también para actualizar su estado. Los datos los tendremos habitualmente en una base de datos, por lo que en los modelos tendremos todas las funciones que accederán a las tablas. Se tienen los siguientes modelos:

- **modelLogin.**
- **modelPonencias.**
- **modelPonentes.**
- **modelAdministrador.**
- **modelParticipante.**
- **modeloPrograma.**

Paquete de controlador

Contiene la parte administrativa del sistema.

- **controladorLogin:** Permite al usuario en general iniciar sesión.
- **controladorPonencias:** El administrador podrá añadir, eliminar y editar las diversas ponencias que se ofrezcan.
- **controladorAdministrador:** El administrador podrá añadir, eliminar y editar los

programas.

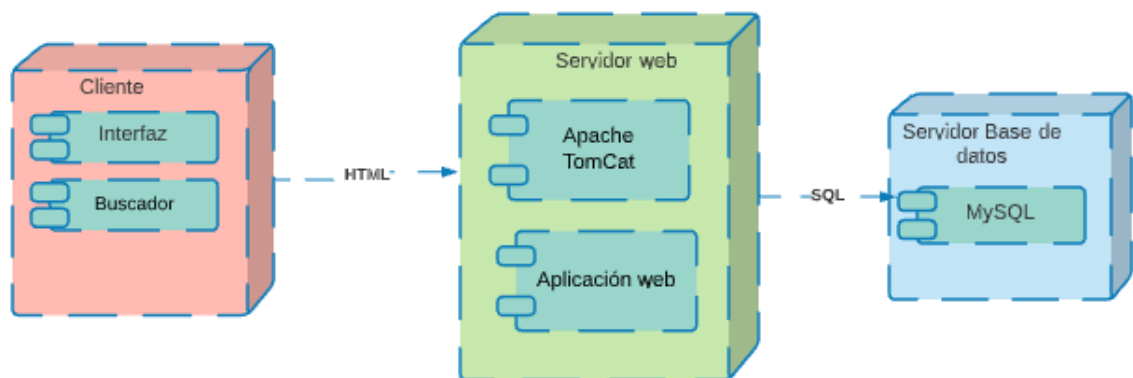
- **controladorPonentes:** Permite al administrador registrar un ponente.
- **controladorParticipante:** Permite al usuario registrarse como participante de un evento.
- **controladorPrograma:** El administrador podrá añadir, eliminar y editar los programas

Base de datos:

Contiene la base de datos que incluye toda la información del sistema.

6. Visión de implantación

El sistema será implantado cuando sea validado por los stakeholders.



En el diagrama, las máquinas físicas se representan como nodos. El nodo es un elemento donde se ejecutan los componentes, así mismo los componentes embebidos son aplicaciones, librerías y motor de base de datos. Ambos se comunicarán a través de relaciones lo que indica el tipo de conexión entre ellos. Toda la aplicación estará montada en un único servidor en el cual se encontrará procesando el servidor de aplicaciones (**Tomcat**) y la base de datos (MySQL).

7. Visión de implementación

El sistema será implementado utilizando conceptos de programación orientada a objetos, modelo MVC, utilizando los lenguajes y tecnologías HTML, JAVA, CSS, JavaScript.

Cliente:

Sistema Operativo	Multiplataforma
Browser	Chrome, Microsoft Edge, Firefox
Tecnologías Utilizadas	<ul style="list-style-type: none">• HTML5• CSS3• jQuery
Lenguajes Utilizados	JavaScript

Servidor de la aplicación:

Sistema Operativo	Multiplataforma
--------------------------	-----------------

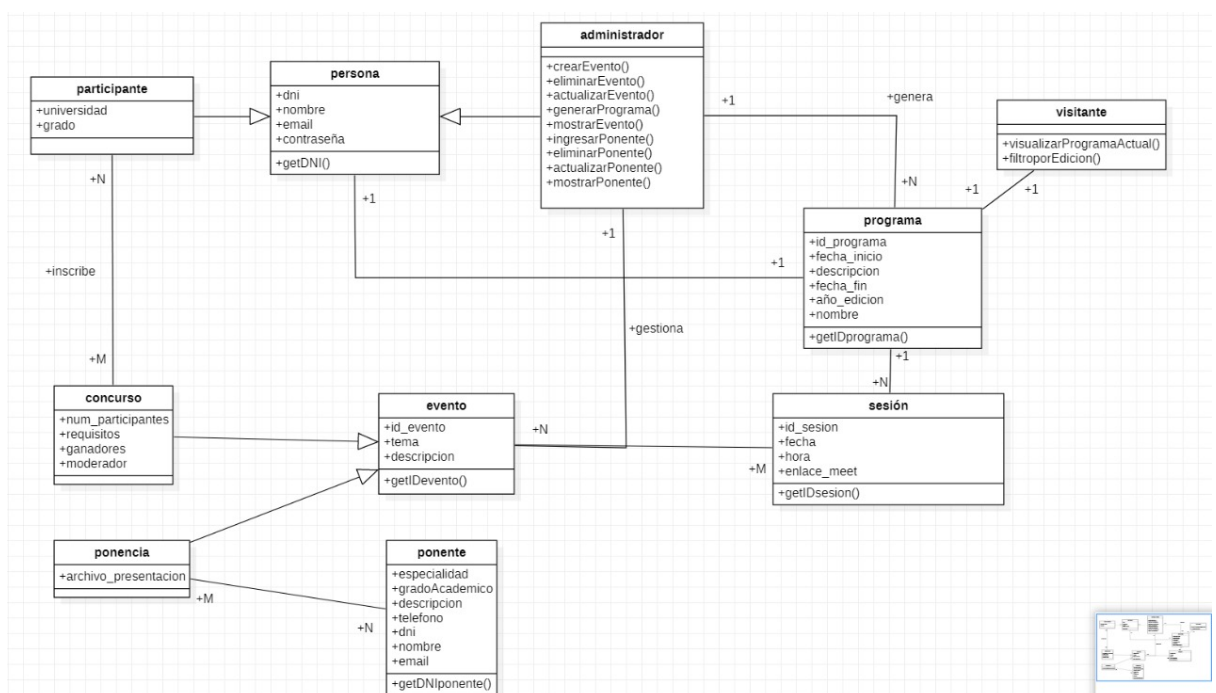
Servidor de la aplicación	Tomcat
----------------------------------	--------

Servidor de la base de datos:

Sistema Operativo	Multiplataforma
Lenguajes Utilizados	mySQL 8.0

7.1. Visión de Datos

El siguiente diagrama ilustra la estructura de datos estáticos y las relaciones de las principales entidades que serán almacenadas por la aplicación en la base de datos.



Resumen: Modelo de Datos usando Diagrama de Clases, del Sistema de Publicación Proceedings de Eventos Semana de la Computación

persona	
dni	int(8)
nombre	varchar
contraseña	varchar
email	varchar
Métodos	
getDNI()	

participante: persona	
universidad	varchar
grado	int

administrador: persona
Métodos:
crearEvento()
eliminarEvento()
actualizarEvento()
generar_programa()
mostrarEvento()
ingresarPonente()
eliminarPonente()
actualizarPonente()
mostrarPonente()

programa	
id_programa	int
fecha_inicio	date
descripcion	varchar
fecha_fin	date
anio_edicion	int(4)
nombre	varchar
Método	
getIDprograma()	

visitante
Método
visualizarProgramaActual()

filtrarPorEdicion()

sesion	
id_sesion	int
fecha	date
hora	date
enlace_meet	text
Método	
getIDsesion()	

evento	
id_evento	int
tema	varchar
descripción	varchar
Métodos	
getIDevento()	

evento: concurso	
num_participantes	int
requisitos	varchar
ganadores	varchar
moderador	varchar

evento: ponencia	
archivo_presentación	varchar

ponente	
especialidad	varchar
gradoAcademico	varchar

descripción	date
teléfono	date
dni	int(8)
nombre	varchar
email	varchar
Métodos	
getDNIponente()	

8. Desempeño

El Sistema de Publicación de Proceedings de Eventos será ejecutado accediendo a su sitio web a través de un navegador web, esta aplicación estará disponible para versión de escritorio.

El desempeño que debe tener el Sistema de Publicación de Proceedings de Eventos se describen en los requisitos a continuación, están relacionados con la velocidad del sistema y la retroalimentación al usuario para que el usuario final no espere demasiado para obtener sus respuestas.

Requisito	Descripción
Tiempo de respuesta	El Sistema debe tener un tiempo de respuesta de una media de un segundo.
Capacidad	Las interfaces del sistema deben de cargar en menos de 3 segundos cuando el número de usuarios simultáneos es mayor a 10 000.
Actualización de datos	Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 2 segundos.
Ingreso simultáneo	El sistema debe ser capaz de realizar validaciones del usuario, soportar el acceso de múltiples usuarios y sus consultas con respecto a la información simultáneamente sin reducir su rendimiento.
Tasa de transferencia	Los procesos del sistema no deben sobrepasar los 5 segundos para cualquier acción.

9. Calidad

Requisito	Descripción
MVC	El sistema debe desarrollarse en la arquitectura con las capas Modelo, Vista y Controlador.
Lenguajes	HTML, JAVA, CSS, JavaScript
Base de Datos	El sistema debe ser modelado para una base MYSQL
IDE	Eclipse

Enlace de repositorio github

<https://github.com/MaryoriHilares/Proyecto-Ing.-de-Software>

Enlace de Trello

<https://trello.com/b/f37ghSe8/sistema-de-publicaci%C3%B3n-de-proceedings-de-eventos>