DOCUMENTO DE ARQUITECTURA

PROYECTO

“Sistema de reconocimiento facial para la identificación de postulantes en el acceso de resultados del examen de admisión de UNMSM”

Grupo #6

**Integrantes:**

1. ALMONACID PARIPANCCA, Antony Brayan
2. MONTALVO GARCIA, Antony Abel
3. TOVAR TABOADA, Ricardo Manuel
4. UNOCC SIHUINTA, Roberto Carlos
5. PERNIA MEZA, Iver Elvis
6. GARCIA MARTINEZ, Christian Arturo

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

E.A.P. INGENIERÍA DE SOFTWARE

FACE SOLUTIONS

**Sistema de reconocimiento facial para la identificación de postulantes en el acceso de resultados del examen de admisión UNMSM.**

***Documento de la Arquitectura de Software***

***Versión 1.0***

**Historial de Revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 06/07/2020 | 1.0 | Creación del documento | -Almonacid Paripancca Antony. |
| 07/07/2020 | 1.1 | Actualización del documento |  |
| 07/07/2020 | 2.0 | Revisión del documento |  |
| 07/07/2020 | 2.1 | Correcciones del documento |  |

**Índice**

1. [Objetivo de solución 5](#_bookmark0)
2. [Definiciones y convenciones 5](#_bookmark1)
3. [Antecedentes 8](#_bookmark2)
4. [Requerimientos 9](#_bookmark3)
   1. [Requerimientos funcionales 9](#_bookmark4)
   2. [Requerimientos no funcionales 11](#_bookmark5)
   3. [Escenarios de tributos de calidad principales 12](#_bookmark6)
5. [Arquitectura general de la solución 14](#_bookmark7)
   1. [Arquitectura de la solución 14](#_bookmark8)
   2. [Tecnologías utilizadas 15](#_bookmark9)
6. [Vistas de la solución 16](#_bookmark10)
   1. [Vista Modular 16](#_bookmark11)

[Vista Modular 16](#_bookmark12)

* 1. [Vista funcional 17](#_bookmark13)

[Diagrama de Caso de Uso 17](#_bookmark14)

* 1. [Vista de componentes 20](#_bookmark15)

[Diagrama de Componentes 20](#_bookmark16)

* 1. [Vista de despliegue 22](#_bookmark17)

[Diagrama de Despliegue 22](#_bookmark18)

* 1. [Vista de integración 23](#_bookmark19)

1. [Evaluación de la arquitectura 24](#_bookmark20)
   1. [Fortalezas 24](#_bookmark21)
   2. [Recomendaciones 24](#_bookmark22)

# Objetivo de solución

Solución: El objetivo de la implementación del sistema propuesto consiste en poder mejorar el sistema de acceso a los resultados del examen de admisión, proponiendo la verificación del postulante a través de un sistema de reconocimiento facial que tenga la capacidad de autenticar y autorizar la identidad de este con respecto a la información de los resultados de la prueba de examen de admisión asociado. Seguidamente, de proporcionar la posibilidad de solicitar el trámite de constancia de ingreso digital.

Para ello el sistema de reconocimiento facial propuesto debe tener en cuenta las siguientes consideraciones para la correcta implementación de este en el proceso de examen de admisión.

* + Realizar coordinaciones con las oficinas correspondientes de UNMSM, con respecto al valor legal de los requisitos de inscripción solicitados para el postulante en el presente sistema propuesto.
  + Gestionar las firmas de las personas encargadas del proceso de emisión de constancias de ingreso digital.
  + Evaluar el desempeño de los empleados, con respecto al uso del sistema.
  + Detectar las necesidades formativas del personal y ofrecer capacitaciones de formación con respecto al uso del sistema.

# Definiciones y convenciones

* FRSIAAATR:Sistema de reconocimiento facial para la identificación de postulantes en el acceso de resultados del examen de admisión de UNMSM.
* FR: Reconocimiento facial.
* ERS: Especificación de requisitos de software.
* Servidor: Computadora conectada a una red que pone sus recursos a disposición del resto de los integrantes de la red. Suele utilizarse para mantener datos centralizados o para gestionar recursos compartidos.
* SGBDR: Sistema de gestión de base de datos relacionales.
* Base de datos: Cualquier conjunto de datos organizados para su almacenamiento en la memoria de un ordenador o computadora.
* Conexión: En referencia a la comunicación existente entre dos entes que tienen características similares para el intercambio de información.
* XML-RPC: Es un protocolo de llamada a procedimiento remoto que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes.
* HTTP: Significa HyperText Transfer Protocol (Protocolo de transferencia de hipertexto) es el método más común de intercambio de información en la World Wide web, el método mediante el cual se transfieren las páginas web a un ordenador.
* SQL: Significa Structured Query Language, un lenguaje de programación estándar e interactiva para la obtención de información desde una base de datos y para actualizarla.
* VM: Significa “máquina virtual”, es una implementación de software de un ambiente de computación en el que se puede instalar y ejecutar un sistema operativo o programa.
* UI: También llamado “User Interface”, se refiere a la creación de la interfaz, que puede ser gráfica o desarrollada por tecnologías como HTML, CSS, jQuery, etc.
* CSS: Su siglas significa “Cascading Style Sheets ” (Hojas de estilo en cascada), es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado, es decir, establece el diseño visual de los documentos web, e interfaces de usuario escritas en HTML.
* Python: Es un lenguaje de programación [interpretado](https://es.wikipedia.org/wiki/Int%C3%A9rprete_(inform%C3%A1tica)) cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.
* JQuery: Es software libre y de código abierto, permitiendo su uso en proyectos libres y privados.
* HTML: Sus siglas significa “HyperText Markup Language” (lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web.

# Antecedentes

Hasta la presente fecha, el proceso de admisión patrocinado año tras año por la Universidad Nacional Mayor de San Marcos es reconocido por ser un evento masivamente activo, es decir, participa un gran porcentaje de postulantes con respecto a otros procesos de admisión hacia otras entidades, por ende, la gestión de estos es un factor de riesgo para el proceso de admisión, además de que posterior a cumplir con los lineamientos solicitados para ser considerado un ingresante, existe un proceso de negocio enfocado en la emisión de constancias que acrediten y certifiquen el ingreso del postulante a una especialidad de la casa de estudios que representa San Marcos.

En la “nueva normalidad”, originada por la pandemia del COVID - 19, se necesita implementar una solución que permita el monitoreo y control de los requisitos presentados por los postulantes previo examen de admisión, además de establecer u flujo de proceso que permita la gestión de constancias de ingreso digital, permitiendo realizar el trámite sin la necesidad de acercarse en persona a las instalaciones de la Universidad Nacional Mayor de San Marcos, siendo dificultoso este último punto en la actualidad por la coyuntura postpandemia en términos de salud.

Así que en el siguiente documento arquitectónico se detallará acerca del Sistema de reconocimiento facial para la identificación de postulantes en el acceso de resultados del examen de admisión de UNMSM, buscando los requisitos funcionales, vistas de negocios, de los componentes desarrollados en el software y una arquitectura propuesta como una posible solución ante esta problemática de día a día.

# Requerimientos

## Requerimientos funcionales

|  |  |
| --- | --- |
| Código | Descripción |
| RF - 01 | El sistema debe permitir al usuario “postulante” registrar los requisitos solicitados para la inscripción al proceso de examen de admisión. |
| RF – 02 | El sistema debe permitir al usuario “postulante” acceder al módulo de postulante, a través de un mecanismo de reconocimiento facial. |
| RF – 03 | El sistema debe permitir al usuario “postulante” visualizar la información relacionada a su perfil. (Datos personales, fotos registradas, bandeja de mensajes). |
| RF – 04 | El sistema debe permitir al usuario “postulante” actualizar los requisitos previamente inscritos por este, en caso haya motivo de observación. |
| RF – 05 | El sistema debe permitir registrar una cuenta de usuario “administrador” para el acceso al módulo de administrador. |
| RF – 06 | El sistema debe permitir al usuario “administrador” visualizar la información relacionada a su perfil. (Datos personales, fotos registradas, bandeja de notificaciones). |
| RF – 07 | El sistema debe permitir al usuario “administrador” buscar los requisitos de un determinado postulante, utilizando filtros como facultad y especialidad. |
| RF – 08 | El sistema debe permitir al usuario “administrador” generar una notificación con respecto a la legibilidad de un requisito(s) presentado por un postulante, en caso haya motivo de observación. |
| RF – 09 | El sistema debe permitir al usuario “postulante”, enviar una solicitud de tramite de constancia de ingreso digital. |
| RF – 10 | El sistema debe permitir al usuario “administrador”, recepcionar las solicitudes de constancia de ingreso digital en una bandeja de trabajo. |

|  |  |
| --- | --- |
| RF - 11 | El sistema debe permitir al usuario “administrador” buscar una solicitud de constancia de ingreso digital en específico, utilizando filtros como facultad, especialidad, rango de fechas. |
| RF – 12 | El sistema debe permitir al usuario “administrador” atender una solicitud de trámite de constancia de ingreso, generando el documento con extensión pdf asociado a la información ligada a la solicitud. |
| RF – 13 | El sistema debe permitir al usuario “administrador” firmar y enviar la solicitud de constancia de ingreso digital, permitiendo añadir la firma (en formato png) de la autoridad respectiva sobre la constancia de ingreso digital pdf y enviarla al correo electrónico del postulante. |
| RF – 14 | El sistema debe permitir al usuario “administrador” visualizar un historial de las solicitudes de constancia de ingreso digital que ha procesado, apoyándose de filtros como facultad, especialidad y rango de fechas. |

## Requerimientos no funcionales

|  |  |
| --- | --- |
| Código | Descripción |
| RNF – 01 | El tiempo para cargar la data no debe ser mayor de 5 segundos. |
| RNF – 02 | Se debe optimizar el tamaño de los documentos de los registrados de cada  trabajador no mayor a 1 Mb. |
| RNF – 03 | El sistema de contar con un manejo de errores frente a eventos no planificados como errores de llenado de datos. |
| RNF – 04 | Las contraseñas deben ser encriptadas antes de ser almacenadas. |
| RNF – 05 | Las contraseñas deberán cumplir un nivel de complejidad tener 8 caracteres como mínimo y usar mayúsculas y minúsculas. |
| RNF – 06 | La interfaz gráfica para el administrador de RR. HH debe ser entendible ante las funciones que realiza. |
| RNF – 07 | No se modificará el código de los frameworks para que la aplicación soporte actualizaciones a nuevas versiones. |
| RNF – 08 | El sistema permitirá cambiar su estilo a través de hojas de estilo en cascada (CSS), que deberán personalizarse para cada componente, añadir nuevos estilos y extender componentes para la reutilización de funcionalidad. |
| RNF – 09 | El sistema cuenta con ayudas para sus funciones básicas. |

## Escenarios de atributos de calidad principales

|  |  |
| --- | --- |
| Escenario de Performance | |
| Fuente | Actualización de información |
| Estímulo | Actividad en el módulo de RR.HH. |
| Entorno | Explotación |
| Artefacto | Servidor central |
| Respuesta | Se registra los cambios y la interacción con la nueva información en un tiempo menor de 3 segundos. |
| Medida | El tiempo que fue modificado |
| Atributo | Perfomance |

|  |  |
| --- | --- |
| Escenario de Disponibilidad | |
| Fuente | Ingreso de documentos |
| Estímulo | Actividad en el módulo de RR.HH. |
| Entorno | Explotación |
| Artefacto | Servidor Central |
| Respuesta | Visualización e interacción con la agregación de nuevos documentos. |
| Medida | Recibir el documento a las 24 horas al día. |
| Atributo | Disponibilidad |

|  |  |
| --- | --- |
| Escenario de Fiabilidad | |
| Fuente | Administrador del área de RR.HH. |
| Estímulo | Generar contratos |
| Entorno | Explotación |
| Artefacto | Servidor Central |
| Respuesta | Se realizará la operación de generar contratos sin fallos. |

|  |  |
| --- | --- |
| Medida | El administrador del área de RR.HH. no debe abandonar la operación en el tiempo de espera. |
| Atributo | Fiabilidad |

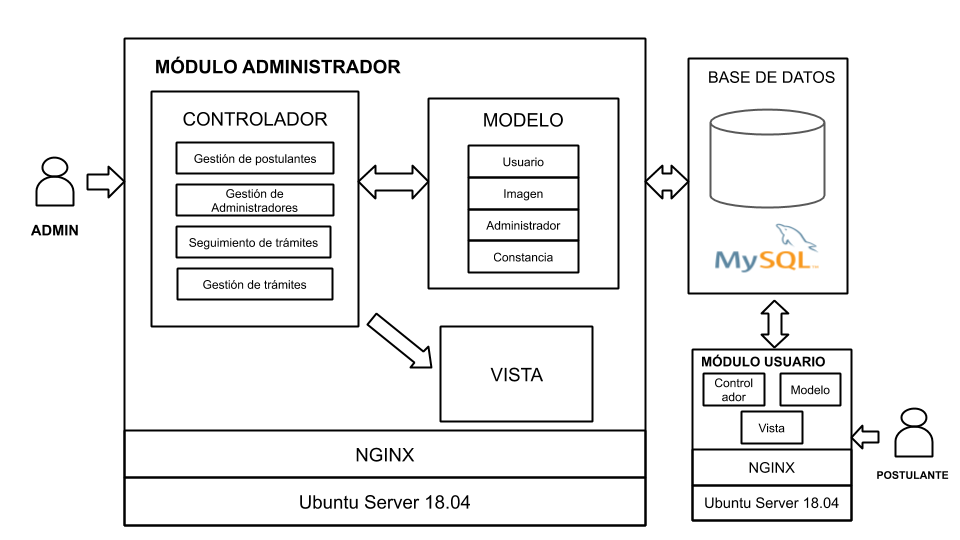
|  |  |
| --- | --- |
| Escenario de Usabilidad | |
| Fuente | Administrador del área de RR.HH. |
| Estímulo | Usar el módulo |
| Entorno | Sistema finalizado y operativo |
| Artefacto | Subsistema del módulo de RR.HH. |
| Respuesta | El subsistema presentará una estructura simple y fácil de utilizar |
| Medida | El administrador del área de RR.HH. debe aprender a utilizar el subsistema en menos de una semana en hora laboral. |
| Atributo | Usabilidad |

|  |  |
| --- | --- |
| Escenario de Confiabilidad | |
| Fuente | Módulos anexados al área de RR.HH. |
| Estímulo | Acceso a la data |
| Entorno | Explotación |
| Artefacto | Servidor Central |
| Respuesta | Aumento de la seguridad del servidor desplegado. |
| Medida | Número de malware igual a cero. |
| Atributo | Confiabilidad |

# Arquitectura general de la solución

## Arquitectura de la solución

En este apartado, se presentará la arquitectura del Sistema de reconocimiento facial para la identificación de postulantes en el acceso de resultados del examen de admisión UNMSM, con el que interactuarán los usuarios y administradores:

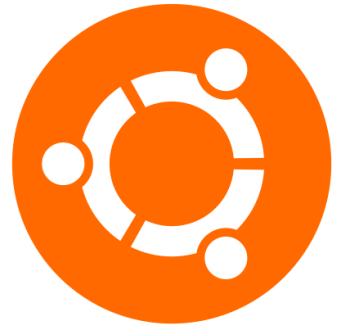


***Diagrama de la Arquitectura Propuesta***

## Tecnologías utilizadas

En esta sección, se especificará las herramientas que se utilizaron para el desarrollo del Sistema de reconocimiento facial para la identificación de postulantes en el acceso de resultados del examen de admisión UNMSM:

* + - **Node.js: Es** un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación [JavaScript](https://es.wikipedia.org/wiki/JavaScript), asíncrono.
    - **MySQL Workbench 8.0:** Sistema de gestión de bases de datos relacional orientado a objetos y de código abierto.
    - **Nginx:** es un servidor web que también actúa como proxy y balanceador de carga. La estructura del software es asíncrona y controlada por eventos; lo cual permite el procesamiento de muchas solicitudes al mismo tiempo.
    - **Ubuntu Server 18.04.1 LTS:** Sistema operativo de código abierto para computadores, compuesto de múltiple software normalmente distribuido bajo una licencia libre o de código abierto.





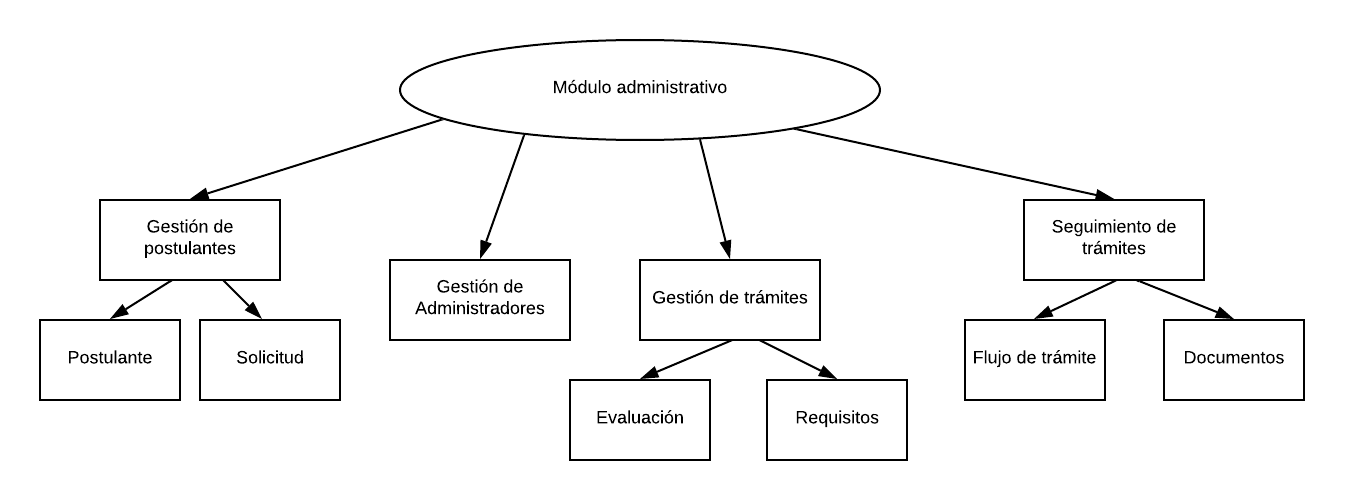
***Figura de Tecnologías Utilizadas***

# Vistas de la solución

## Vista Modular

### Vista Modular

Pretendemos mostrar a través de esta vista la división funcional del proyecto, el cual está dividido en 2 módulos:



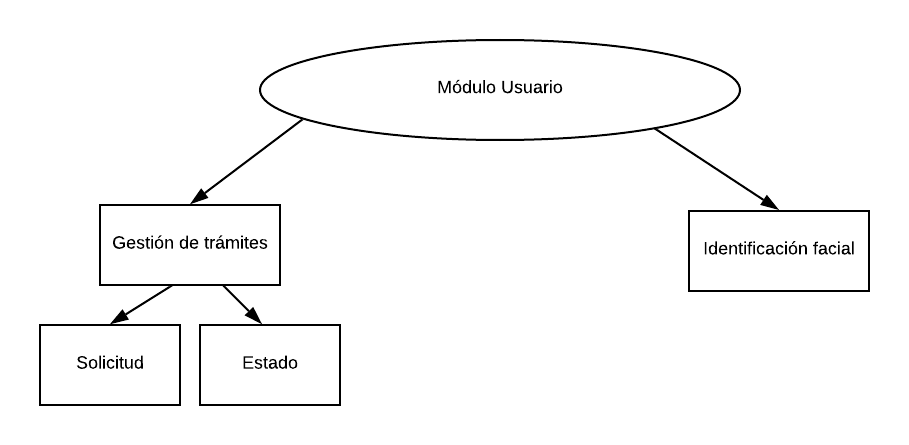
***Diagrama de la Vista Modular (1)***

**Gestión de postulantes**: Sub módulo encargado de listar a los postulantes que han realizado el examen y el estado según su resultado. Además, si su resultado es exitoso puede realizar la solicitud para tramitar su constancia de manera digital.

**Gestión de Administradores**: Sub módulo encargado de mostrar información del administrador actualmente logueado en el sistema y la posibilidad de crear administradores.

**Gestión de trámites**: Sub módulo encargado de controlar los requisitos de las solicitudes de trámite realizadas por los usuarios y evaluar si los requisitos enviados se aprueban o se observan.

**Seguimiento de trámites**: Sub módulo encargado de mostrar el flujo que sigue la solicitud a través de las diferentes entidades de la universidad antes de generar la constancia virtual y mostrar los documentos involucrados en este flujo.

****

***Diagrama de la Vista Modular (2)***

La siguiente vista modular hace referencia al módulo disponible para el postulante, dividiendo así el proyecto en 2:

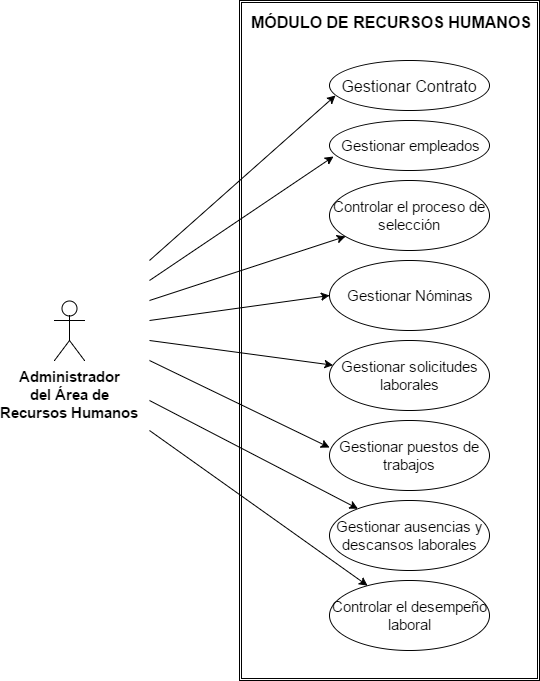
**Gestión de trámites:** El postulante podrá realizar su solicitud para constancia virtual y observar el estado de está.

**Identificación facial:** Para el correcto ingreso al sistema, hará uno de un método de identificación facial para validar su ingreso.

## Vista funcional

### Diagrama de Caso de Uso

El diagrama de Caso de Uso representa al Administrador del Área de Recursos Humanos que puede interactuar con el Módulo de Recursos Humanos y puede hacer las siguientes funciones:



***Diagrama de Caso de Uso***

|  |  |
| --- | --- |
| ID | CUS - 01 |
| Caso de Uso: | Gestionar Contratos |
| Actor: | Administrador del Área de Recursos Humanos |
| Descripción: | El Administrador del Área de Recursos Humanos podrá generar contratos, así como renovar o  finalizar contratos. |

|  |  |
| --- | --- |
| ID | CUS - 02 |
| Caso de Uso: | Gestionar Empleados |
| Actor: | Administrador del Área de Recursos Humanos |
| Descripción: | El Administrador del Área de Recursos Humanos podrá crear y modificar los empleados. |

|  |  |
| --- | --- |
| ID | CUS - 03 |
| Caso de Uso: | Controlar el proceso de selección |
| Actor: | Administrador del Área de Recursos Humanos |
| Descripción: | El Administrador del Área de Recursos Humanos podrá definir, modificar, y monitorear las etapas  del proceso de selección para los puestos de trabajo disponible. |

|  |  |
| --- | --- |
| ID | CUS - 04 |
| Caso de Uso: | Gestión de Nominas |
| Actor: | Administrador del Área de Recursos Humanos |
| Descripción: | El Administrador del Área de Recursos Humanos podrá registrar y realizar el cálculo de los pagos  de empleados. |

|  |  |
| --- | --- |
| ID | CUS - 05 |
| Caso de Uso: | Gestión de Solicitudes Laborales |
| Actor: | Administrador del Área de Recursos Humanos |
| Descripción: | El Administrador del Área de Recursos Humanos podrá registrar y modificar las solicitudes  laborales presentadas en la oficina de RRHH. |

|  |  |
| --- | --- |
| ID | CUS - 06 |
| Caso de uso | Gestionar puestos de trabajos |
| Actor: | Administrador de Recursos Humanos |
| Descripción: | El Administrador del Área de Recursos Humanos podrá registrar y modificar los puestos de  trabajos de cada empleador. |

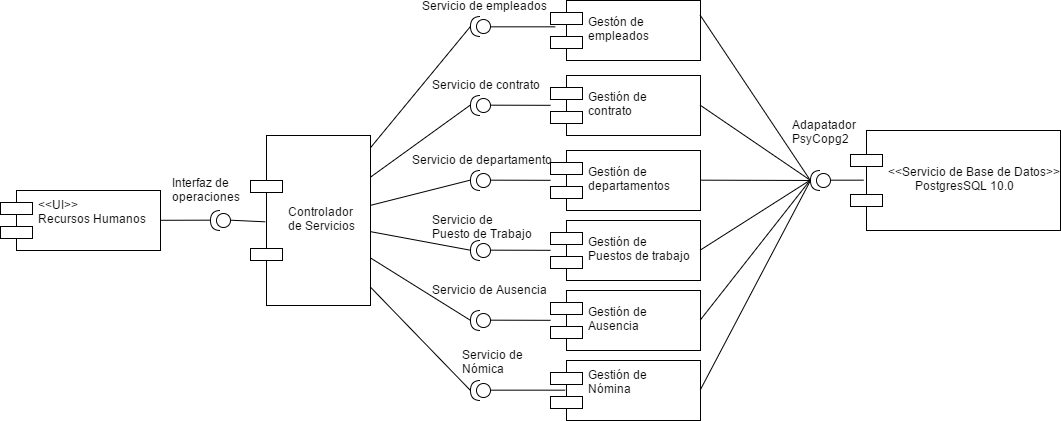
|  |  |
| --- | --- |
| ID | CUS - 07 |
| Caso de uso | Gestionar ausencias y descansos laborales |
| Actor: | Administrador de Recursos Humanos |
| Descripción: | Registrar la no disponibilidad del trabajador ante las horas laborales y los días que sean declarados  no laborales según la empresa. |

|  |  |
| --- | --- |
| ID | CUS - 08 |
| Caso de uso | Controlar el desempeño laboral |
| Actor: | Administrador de Recursos Humanos |
| Descripción: | Calificar el desenvolvimiento de sus actividades dentro de determinadas fechas, siendo alta en  caso que tenga un eficiente capacidad ante sus labores y baja en caso contrario. |

## Vista de componentes

### Diagrama de Componentes

El diagrama de Componentes muestra los módulos en los que estará dividida la funcionalidad del sistema. Los conectores identifican la relación entre los mismos.

La aplicación cuenta con módulos claramente definidos los cuales acceden al modelo de datos único de la solución.

***Diagrama de Componentes***

<<UI>>Recursos Humanos: Interfaz gráfica del entorno del módulo de RR.HH.

Controlado de Servicios: Componente donde se concentra la lógica de negocio ya que referencia las funciones asignadas entre la interfaz con los componentes de las operaciones del módulo de RR.HH.

Gestión de empleados: Componente que realiza la operación de gestionar a los empleados.

Gestión de contratos: Componente que realiza la operación de gestionar los contratos. Gestión de departamentos: Componente que realiza la operación de gestionar a los departamentos.

Gestión de puestos de trabajo: Componente que realiza la operación de gestionar los puestos de trabajos.

Gestión de Ausencias: Componente que realiza la operación de gestionar las ausencias de los empleados.

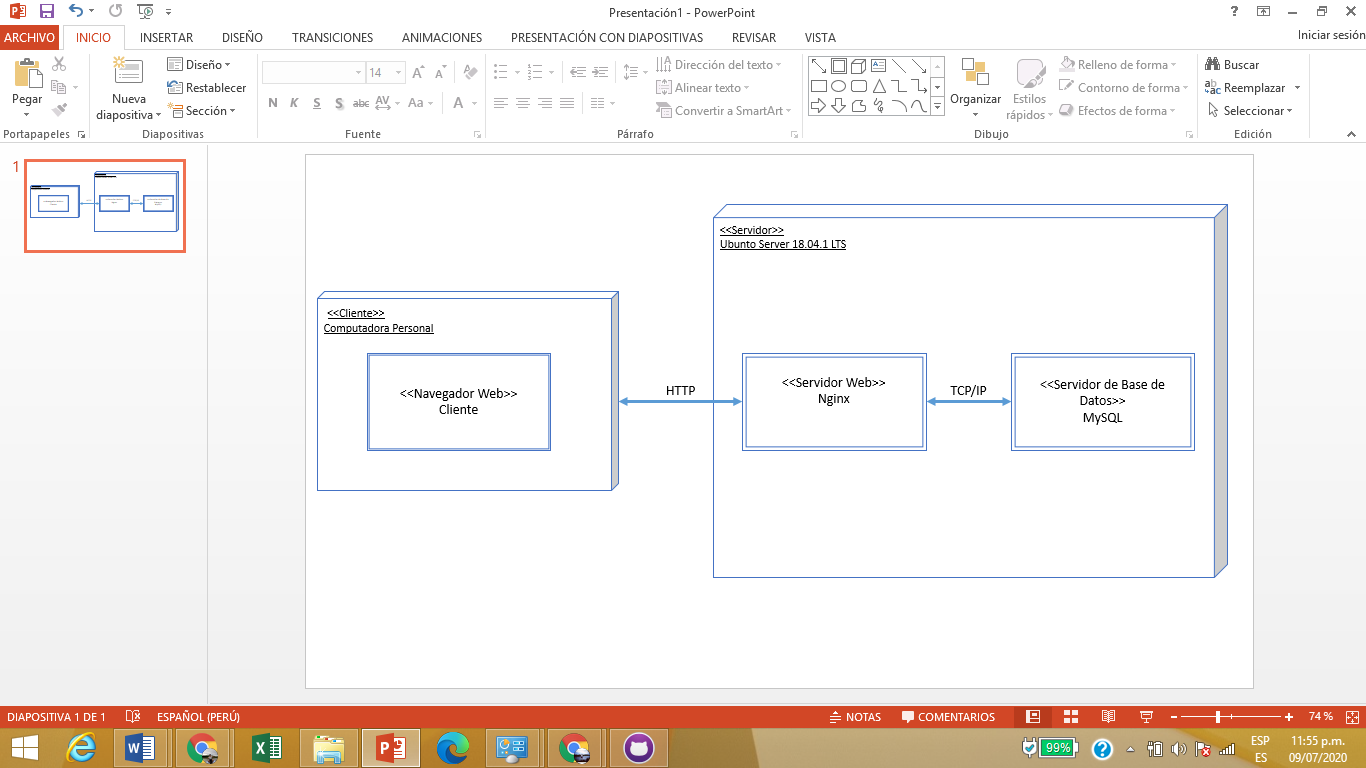
Gestión de Nómina: Componente que realiza la operación de gestionar las nóminas como los pagos hacia los respectivos empleados.

PostgresSQL: Motor de Base de Datos seleccionado para la aplicación.

## Vista de despliegue

### Diagrama de Despliegue

En esta vista se muestra la representación física de la implementación del Sistema, describiendo cada uno de los nodos existentes y como estos se encuentran conectados en la red.



***Diagrama de Despliegue***

* A través de este tipo de PC los distintos usuarios accederán al sistema web, por protocolo HTTP, en una red WAN.
* Se ha usado la plataforma de computación en Nginx, en el que tendremos nuestro Servidor Web. Los usuarios del sistema pueden acceder a los datos por medio del servidor web.
* El Servidor Web se conecta a la Base de Datos mediante TCP/IP, la ruta de conexión lo genera Nginx.
* Se usará la Base de Datos MySQL.
* **Navegador**

Características:

* Chrome y Firefox: año 2018 en adelante
* Otros: últimas 3 versiones
* **Servidor Web**

Debe estar alojado en un servidor web con las siguientes características:

* Procesador mínimo de 4 núcleos, 3 GHz.
* Memoria RAM mínimo de 4GB (6GB como recomendación)
* HDD mínimo de 64GB.
* **Servidor Base de Datos MySQL**

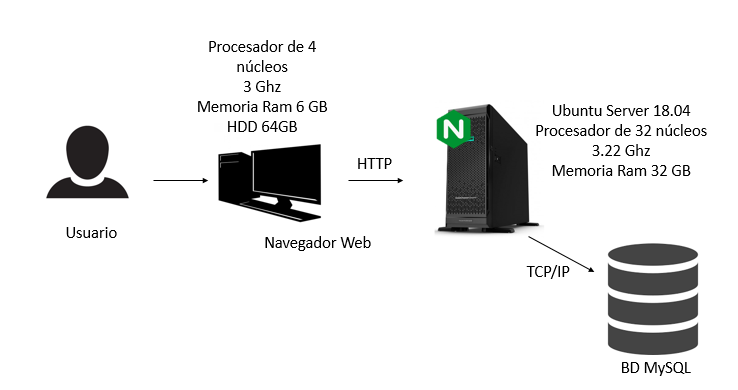
Características:

* Procesador: Intel Xeon E7 2.4 GHz / acceso de memoria de hasta 1066 MHz.
* Memoria RAM 16GB.
* Disco duro SAS 9.6 TB por chasis.

## Vista de integración

* ***Vista de Integración***

En este apartado, veremos que la vista de integración en como grupos de servidores colaboran con el fin de la conexión entre los módulos correspondiente.



***Diagrama de la Vista de Integración***

IBM VM Ubuntu Server 18.04.01 LTS: Servidor del Sistema Operativo donde se desplegará los aplicativos.

BD MySQL: Motor de Base de Datos seleccionado para la aplicación.

# Evaluación de la arquitectura

Ya teniendo esquematizado las arquitecturas funcionales, se definirá acerca de las ventajas que nos brindará con su implementación.

## Fortalezas

El sistema posee las siguientes fortalezas:

* + - Tiene una alta usabilidad, sus interfaces gráficas son muy manejables y entendibles para los administradores y usuarios del sistema.
    - El sistema usa tecnologías bastante reconocidas y confiables, que pueden prestar soporte en caso de algún problema.
    - Al presentar una arquitectura MVC, hace el sistema más escalable ya que al estar separados los componentes entre sí, cada desarrollador puede enfocarse en uno de estos de manera especializada.
    - La arquitectura MVC permite reutilizar los componentes para reducir código repetido y mejorar la calidad del sistema.
    - Sistema divido en dos subsistemas que hacen que cada uno se enfoque en cierto tipo de usuario, dándole una mejor experiencia en el uso.

## Recomendaciones

* + - Cada subsistema depende de un servidor, de lo cual, ante cualquier caída, puede dejar inactivo uno de ellos, sugiriendo a que utilice otros servidores de respaldo.
    - El módulo debería implementar una conexión web service que permita acceder a los datos de los postulantes, que será proporcionado por la Universidad Nacional Mayor de San Marcos.