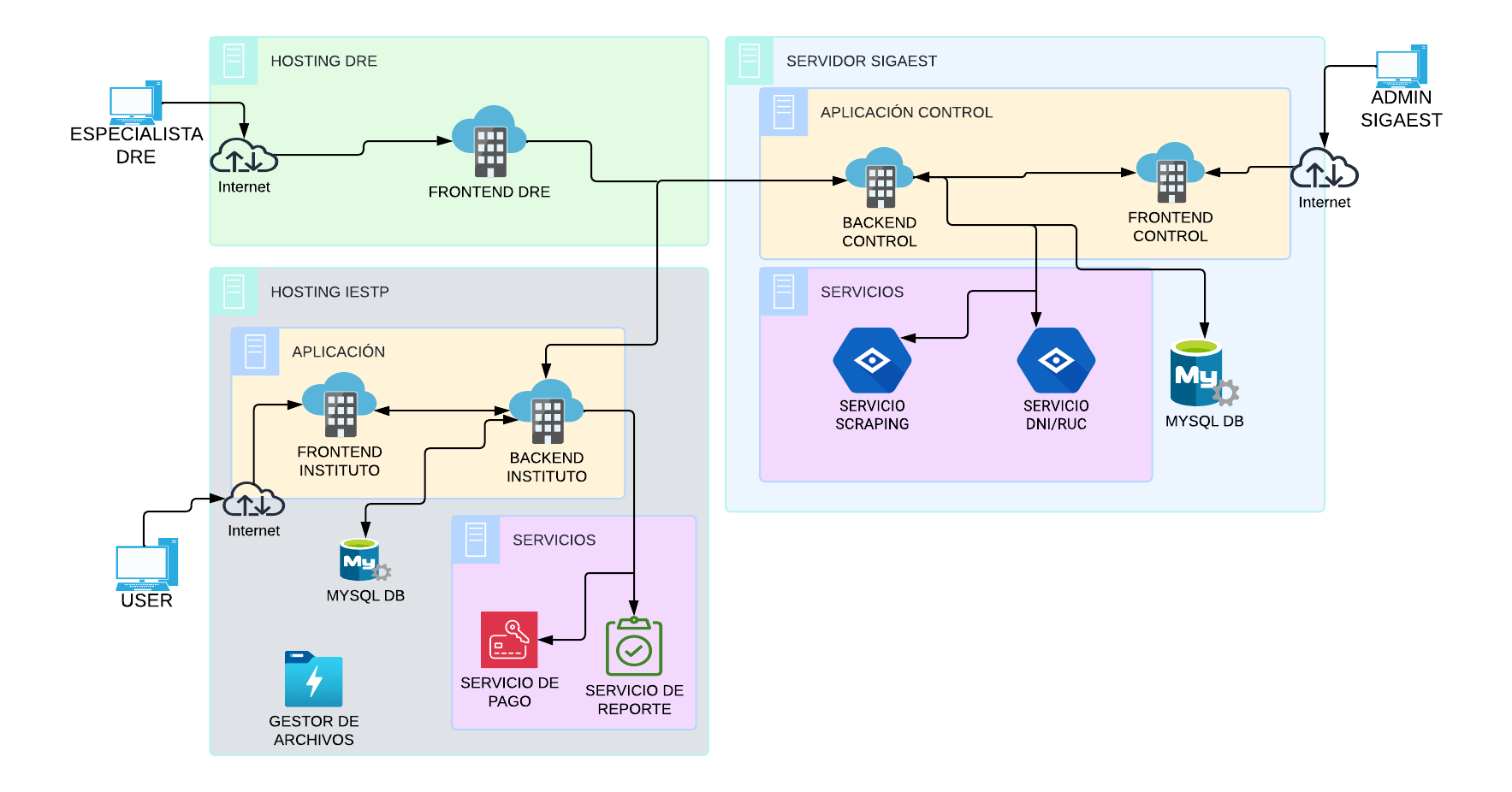


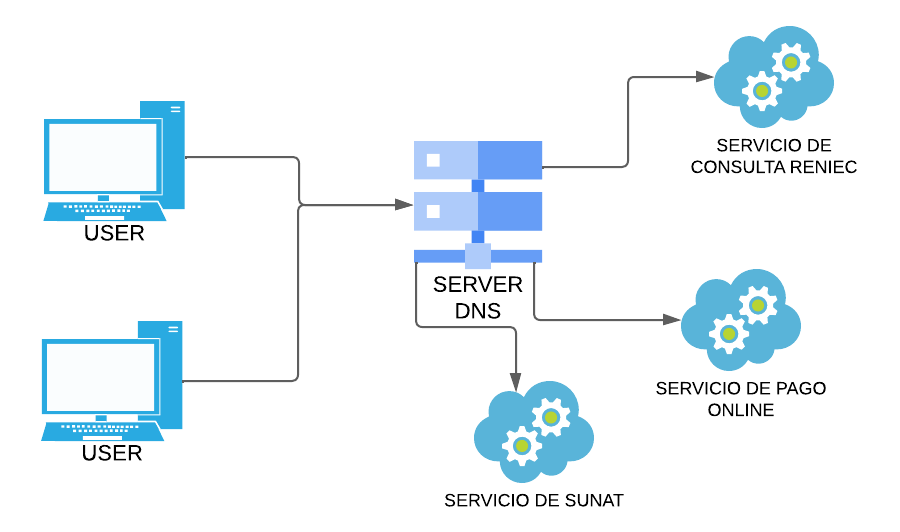
1. **ARQUITECTURA DE TI**
   1. **DISEÑO DE ARQUITECTURA TI**



**DESCRIPCIÓN:**

| **SERVIDOR SIGAEST** | | | | |
| --- | --- | --- | --- | --- |
| **SISTEMAS** | **DESCRIPCIÓN** | **TECNOLOGÍAS** | **ALCANCE** | **OBJETIVO** |
| FRONTEND CONTROL | Este sistema debe ser instalado en el servidor del **SIGAEST**. | * NextJS | Se debe conectar con el **Backend Control** | - |
|
| BACKEND CONTROL | Este sistema debe ser instalado en el servidor del **SIGAEST**.  Este sistema debe validar las instalaciones de clientes institutos y dre mediante un token de aplicación. | * NestJS * MySQL * TypeORM | Se debe conectar con el servicio de **web Scraping** | Obtener información de las ofertas laborales según programa de estudios. |
| Se debe conectar con el servicio de consulta **DNI/RUC** | Obtener información desde la **RENIEC/SUNAT** |
| WEB SCRAPING | El servicio debe ser instalado en el servidor del **SIGAEST.** | * Python | Se debe conectar mínimo a un portal de empleos (Boomerang). | - |
|
| CONSULTA DNI/RUC | Este servicio debe ser implementado en el **Backend Control**. | - | - | - |
| FRONTEND DRE | Este sistema debe ser instalado en el servidor del **SIGAEST**. | * NextJS | Se debe conectar a la **API** del **Backend Control.** | Obtener datos de la institución y su relación con los institutos en su jurisdicción. |
| Se debe conectar al **backend del instituto**. | Obtener estadísticas globales de los institutos en su jurisdicción. |
| **HOSTING INSTITUTO** | | | | |
| **SISTEMAS** | **DESCRIPCIÓN** | **TECNOLOGÍAS** | **ALCANCE** | **OBJETIVO** |
| FRONTEND INSTITUTO | Este sistema debe ser instalado en el hosting de cada instituto. | * NextJS | Se debe conectar con el **Backend del Instituto.** | - |
|
| BACKEND INSTITUTO | Este sistema debe ser instalado en el hosting de cada instituto.  Este sistema debe contar con un token de aplicación. | * NestJS * MySQL * TypeORM | Se debe conectar a la **API** del **Backend Control.** | Para acceder a los servicios de **Web Scraping** y **Consulta DNI/RUC.** |
| Se debe conectar con el servicio de **Reportes.** | Generar reportes en **EXCEL** y **PDF** |
| Se debe conectar con el servicio de **Pagos Online.** | Procesar pagos desde la aplicación. |
| REPORTES | El servicio debe ser instalado en el hosting del instituto**.** | * No definido | El servicio debe aceptar dataset o json para la generación de reportes. | - |
|
| PAGOS ONLINE | Este servicio debe ser implementado en el **Backend del Instituto**. | * Requiere credenciales en el .env | Cada institución debe gestionar la cuenta de pagos. | - |
| GESTOR DE ARCHIVOS | Se debe usar el gestor de archivos del hosting del instituto. |  |  |  |

* 1. **SERVICIOS DE TERCEROS**

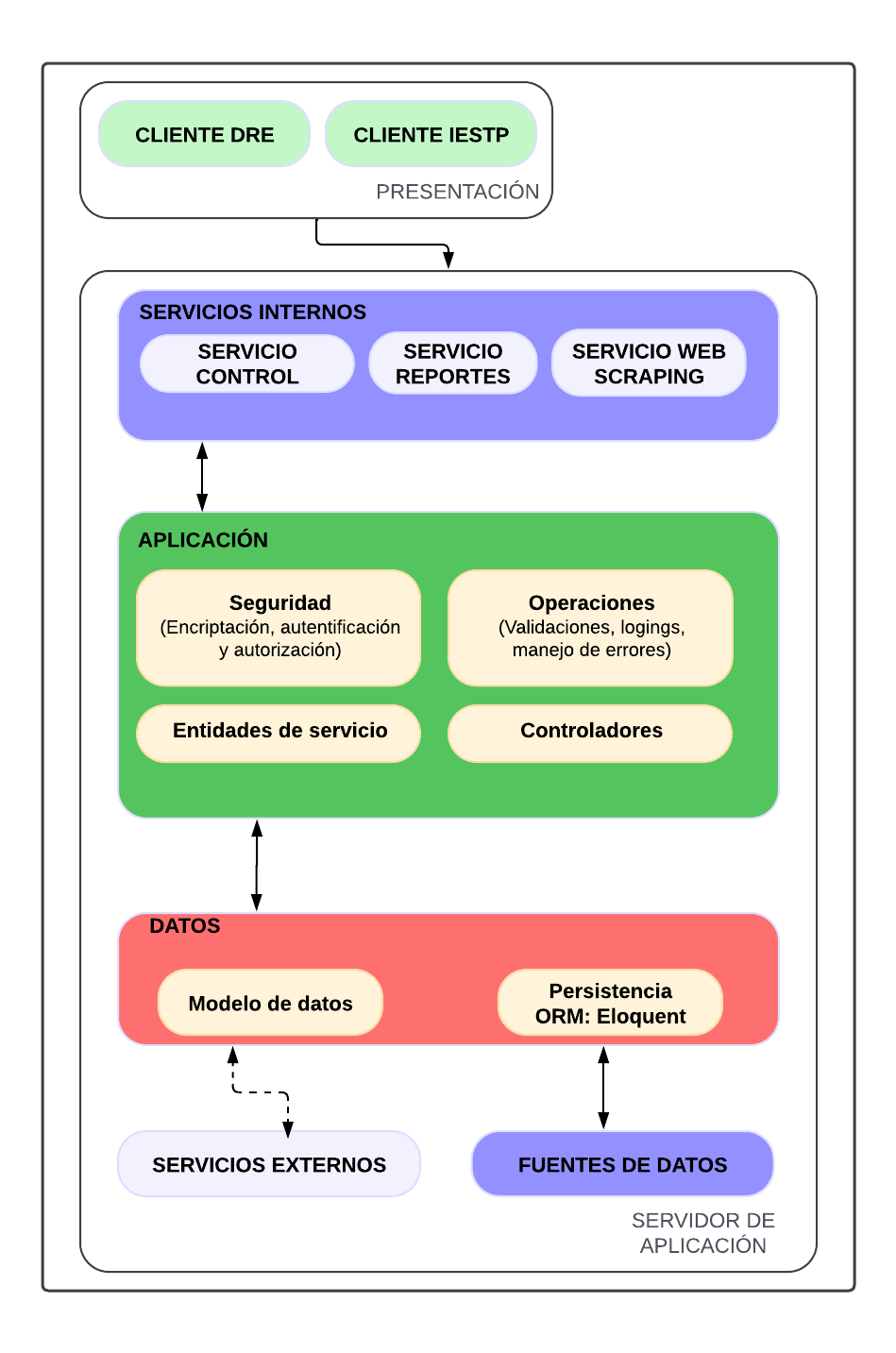


* 1. **HARDWARE Y SOFTWARE**

Se presentan las capacidades de hardware requerido para la implementación.

| **Hardware / Software** | **Característica** | **Descripción** |
| --- | --- | --- |
| **Servidor** | Sistema Operativo | CloudLinux(recomendado) u otra distribución de Linux. |
| Procesador Mínimo | Intel(R) 2.5 GHz |
| Núcleos | 8 |
| Memoria RAM | 16 GB |
| **Almacenamiento** | Almacenamiento requerido | 500GB Carpeta de archivos |
| **Base de datos** | Software requerido | MySQL |
| Tipo | SQL |
| Versión | 8.0 o superior |

1. **ARQUITECTURA DE SOFTWARE**
   1. **DISEÑO DE ARQUITECTURA LÓGICA**



* 1. **DESCRIPCIÓN DE LA ARQUITECTURA**

Se mantiene la arquitectura en cliente servidor.

Por el lado del cliente se va desarrollar usando NextJS, mientras que para el backend NestJS.

* 1. **METAS Y RESTRICCIONES**
     1. **Metas:**
* El sistema debe poder ser accedido a través de una dirección Web pública.
* El sistema debe poder atender las necesidades de concurrencia de acceso y carga.
* La interfaz de usuario debe hacer uso de tecnologías de lado de cliente para el nivel de presentación (web)
* La interfaz de usuario debe ser adaptativa y debe poderse acomodar a distintos dispositivos.
* La arquitectura busca tener bajo acoplamiento entre sus componentes basándose en el principio de única responsabilidad por cada uno de estos.
* El sistema debe permitir un fácil uso para cualquier tipo de usuario, reduciendo así el tiempo en las capacitaciones.
* El sistema debe utilizar la infraestructura de servidores Linux que es actualmente utilizada en la empresa y por lo tanto se deben utilizar tecnologías libres.
  + 1. **Restricciones**
* El sistema deberá ser desarrollado utilizando NextJS en el cliente y Laravel en el servidor para los institutos.
* El sistema deberá soportar los navegadores IE 11+, y las últimas versiones de Mozilla Firefox 4.x y Google Chrome.
* El sistema debe utilizar la infraestructura de servidores en Linux.
* El sistema deberá presentar total conectividad con la Base de Datos.