MANUAL TÉCNICO

SISTEMA DE INFORMACIÓN PARA LA GESTIÓN DE ADMISIONES, EXPEDIENTES, SOLICITUDES ADMINISTRATIVAS, CITAS, REPARACIONES, ESPACIOS Y EMPLEADOS DE LA CORPORACIÓN DE RESIDENCIAS UNIVERSITARIAS.

JEFERSON STEVEN GUEVARA SANCHEZ

INSTITUCION UNIVERSITARIA DE COLOMBIA

FACULTAD INGENIERIA DE SISTEMAS

BOGOTA D.C

2018

Tabla de Contenido

[1. PLAN DE LEVANTAMIENTO DE INFORMACIÓN 4](#_Toc511604503)

[1.1 Métodos de levantamiento de información 4](#_Toc511604504)

[1.2 Observación Directa 5](#_Toc511604505)

[1.3 Entrevista 6](#_Toc511604506)

[1.4. Encuestas 8](#_Toc511604507)

[1.5. Análisis de Documentos 10](#_Toc511604508)

[2. INVESTIGACIÓN DE DEFINICIÓN DE TIPO PRODUCTO 12](#_Toc511604509)

[3. REQUERIMIENTOS 18](#_Toc511604510)

[3.1. Requerimientos funcionales 18](#_Toc511604511)

[3.2. Requerimientos no funcionales 19](#_Toc511604512)

[3.2.1. Motor de Base de Datos 19](#_Toc511604513)

[3.2.2. Software de Desarrollo 19](#_Toc511604514)

[3.2.3. Infraestructura 20](#_Toc511604515)

[3.2.4. Seguridad 20](#_Toc511604516)

[3.2.5. Diseño Corporativo 21](#_Toc511604517)

[3.2.6. Disponibilidad 22](#_Toc511604518)

[4. MODELAMIENTO DE BASE DE DATOS 23](#_Toc511604519)

[4.1. Modelo Relacional 23](#_Toc511604520)

[4.2. Script de Base de Datos 24](#_Toc511604521)

[4.3. Diccionario de Datos 40](#_Toc511604522)

[5. DEFINICION DE ACTORES Y ROLES 49](#_Toc511604523)

[6. CASOS DE USO 51](#_Toc511604524)

[6.1. Caso de Uso Alto Nivel 51](#_Toc511604525)

[6.2. Caso de Uso Extendido 55](#_Toc511604526)

[7. DIAGRAMA DE CLASES 64](#_Toc511604527)

[7.1. Diagrama de Clases 64](#_Toc511604528)

[7.2. Diagrama de Clases de Control 65](#_Toc511604529)

[7.3. Diagrama de Clases de Navegación 66](#_Toc511604530)

[8. DIAGRAMA COMPONENTES 67](#_Toc511604531)

[9. DIAGRAMA DE PAQUETES 68](#_Toc511604532)

LISTA DE ILUSTRACIONES

[Ilustración 1 Corporación de Residencias Universitarias 6](#_Toc511603439)

[Ilustración 2 Flujo de información de la CRU 11](#_Toc511603440)

[Ilustración 3 Diagrama de caso de uso de alto nivel Director 52](#_Toc511603441)

[Ilustración 4 Diagrama de caso de uso de alto nivel Estudiante 53](#_Toc511603442)

[Ilustración 5 Diagrama de caso de uso de alto nivel Psicólogo 54](#_Toc511603443)

[Ilustración 6 Diagrama de caso de uso de alto nivel Mantenimiento 55](#_Toc511603444)

[Ilustración 7 Diagrama de caso de uso de extendido Gestión de Empleado 56](#_Toc511603445)

[Ilustración 8 Diagrama de caso de uso de extendido Gestión de Admisiones 57](#_Toc511603446)

[Ilustración 9 Diagrama de caso de uso de extendido Gestión de Citas Estudiante 58](#_Toc511603447)

[Ilustración 10 Diagrama de caso de uso de extendido Gestión de Citas Psicólogo 59](#_Toc511603448)

[Ilustración 11 Diagrama de caso de uso de extendido Gestión de elementos 60](#_Toc511603449)

[Ilustración 12 Diagrama de caso de uso de extendido Gestión de Espacios 61](#_Toc511603450)

[Ilustración 13 Diagrama de caso de uso de extendido Gestión de Estudiantes 62](#_Toc511603451)

[Ilustración 14 Diagrama de caso de uso de extendido Gestión de Reparaciones 63](#_Toc511603452)

[Ilustración 15 Diagrama de caso de uso de extendido Gestión de Solicitudes 64](#_Toc511603453)

[Ilustración 16 Diagrama de Clases de Control 66](#_Toc511603454)

[Ilustración 17 Diagrama de Clases de Navegación 67](#_Toc511603455)

[Ilustración 18 Diagrama de Componentes 68](#_Toc511603456)

[Ilustración 19 Diagrama de Paquetes 69](#_Toc511603457)

# **PLAN DE LEVANTAMIENTO DE INFORMACIÓN**

A continuación, se describen los pasos realizados para la identificación de procesos, roles involucrados internos y externos, así como las áreas en que se encuentra distribuida la Corporación de Residencias Universitarias.

**Objetivos del levantamiento de información**

* Identificar las áreas en que está compuesta la corporación de residencias universitarias.
* Identificar los procesos que tiene la Corporación de Residencias Universitarias.
* Identificar los Roles que se involucran en los procesos de inicio a fin.
* Describir las herramientas que colaboran en la ejecución de cada uno de los procesos.
  1. **Métodos de levantamiento de información**

**Observación Directa**: Labor de campo donde se realizará un conocimiento de las instalaciones físicas por áreas de la Corporación de Residencias universitarias

**Entrevista**: Se realizará una entrevista al responsable de cada área con el fin de identificar los procesos, documentos, formatos que se manejen.

**Encuesta**: Con los datos obtenidos de la entrevista se realizará una serie de preguntas a los involucrados en cada área, con el fin de cuantificar los datos y poder así obtener un análisis más cerrado.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fase | Actividades | Fecha | | Responsable | Recursos |
| Inicio | Fin |
| Levantamiento de información | Realizar entrevistas | 2 de Enero de 2017 | 5 de Enero de 2017 | Jefersson Steven Guevara | Formato de Entrevista |
| Aplicar Encuestas | 6 de Enero de 2017 | 10 de Enero de 2017 | Jefersson Steven Guevara | Formato  Encuestas |
| Analizar documentos internos | 11 de Enero de 2017 | 15 de Enero de 2017 | Jefersson Steven Guevara | Documentos y formatos internos |
| Observar Procesos y procedimientos | 16 de Enero de 2017 | 20 de Enero de 2017 | Jefersson Steven Guevara | Visita en sitio |

Tabla 1 Plan de levantamiento de información

Para tal fin se emplearon cuatro (4) tipos de procedimientos, a través de igual número de artefactos, que permitieron recopilar información para analizar y viabilizará técnica, económica y administrativamente el proyecto.

* 1. **Observación Directa**

En la visita a las instalaciones de la Corporación de residencias universitarias, ubicadas en la Cra 33 No 23 – 87, se identificaron 2 bloques de 13 pisos, cada piso con 9 habitaciones. Donde en el primer piso se encuentra ubicada la administración, compuesta por el director, el psicólogo, y mantenimiento.



Ilustración 1 Corporación de Residencias Universitarias

En este primer piso se encuentra el área de archivo, así como el espacio de atención psicológica, el cual permite atender a los estudiantes cuando lo necesiten; también la recepción de los daños que se presenten en los diferentes apartamentos. Además, está ubicada el área de admisiones, donde se realiza la recepción de los formatos y documentos para concursar por un cupo en las instalaciones.

* 1. **Entrevista**

Con el fin de identificar cada una de las áreas y sus procesos, se realizó una entrevista a cada uno de los responsables, incluyendo los estudiantes quienes son el cliente principal.

|  |  |
| --- | --- |
| **Formato de entrevista para el levantamiento de información de los procesos de la CORPORACION DE RESIDENCIAS UNIVERSITARIAS** | |
| **Fecha** | |
| **Área** | |
| **Nombre del Responsable** | |
| **Cargo** | |
|  |  |
| **1.** | Que procesos realiza en esta área? |
| **2.** | Como se realizan tales procesos? |
| **3.** | Con que frecuencia se realizan estas procesos? |
| **4.** | Que decisiones toma en estos procesos? |
| **5.** | Tiene algún tipo de restricción de tiempo en la ejecución de estos procesos? |
| **6.** | Lleva registro de estas restricciones? |
| **7.** | Que impacto tiene la ejecución de estos procesos con otras áreas? |
| **8.** | Depende de aprobaciones externas para la ejecución de tales procesos? |
| **9.** | Notifica a otras áreas la ejecución de estos procesos, por qué medio? |
| **10.** | Que herramientas informáticas utiliza para el desarrollo de los procesos? |

Tabla 2 Formato de entrevista

Con este catálogo de preguntas se logró establecer que la CRU cuenta con seis (6) procesos centrales para el desarrollo de sus actividades:

**Proceso de admisiones**: Consiste en llevar el control de los estudiantes interesados en recibir un cupo en la CRU, donde se registran por medio de un formato que se entrega a los interesados, después entregan los siguientes documentos: formato de admisión, certificado de notas de la universidad, fotocopia del documento de identidad, certificado de salud de la universidad, certificación laboral de los padres del estudiante. Si cumple con la documentación, es citados a una entrevista y conocen el resultado, por medio de la publicación de un listado en la portería.

**Proceso de gestión de expedientes:** Consiste en registrar, almacenar, consultar y actualizar la información de cada estudiante, presentada en el proceso de admisión. Adicional a esto se puede consultar los datos del acudiente y la habitación asignada.

**Proceso de espacios:** Consiste en el control de cupos de todos los espacios universitarios, donde se puedan cambiar estudiantes entre los espacios. También se podrá consultar la ficha técnica de cada espacio (capacidad, elementos asignados), además de las reparaciones que haya tenido.

**Proceso de control de inventario:** Consiste en el control de activos desde su ingreso a la bodega hasta su asignación en los diferentes espacios.

**Proceso de solicitudes administrativas:** Consiste en registrar las solicitudes administrativas y asignarlas a los roles disponibles:

* Director: recibirá todas las solicitudes y asignará de acuerdo a la categoría, atenderá solicitudes administrativas
* Psicólogo: recibirá las solicitudes de citas, y administrativas como lo requiera el director, para dar solución.

De acuerdo a la categoría, prioridad, se llevará un control del tiempo de solución, por otra parte, permitirá hacer un análisis de las partes que conlleven a dar la solución, puesto que puede tener involucrado al director, al psicólogo.

**Proceso de Atención de Citas:** Consiste en la asignación de citas con el psicólogo, para los estudiantes, y así llevar un registro de las intervenciones psicológicas que se realicen. Para el rol de psicólogo, y director, se permite la consulta y la documentación de las mediaciones realizadas, para así dar cierre a la solicitud asociada, las cuales se asocian al expediente.

**Proceso de Reparaciones:** Cosiste en la gestión de las reparaciones que se generan y a su vez documentar las tareas que fueron necesarias para terminar la reparación, incluyendo su costo.

**Proceso de Gestión de Empleados**: Consiste en administrar los empleados que disponga la CRU, para cada una de las áreas, con el fin de registrar información de contacto, así como los datos de ingreso al sistema y los datos de un contrato básico.

## **Encuestas**

Con el propósito de identificar la situación actual, los inconvenientes existentes y encontrar las oportunidades de mejora, se emplearon tres tipos de artefactos que permitieron viabilizar técnica, económica y administrativamente el proyecto:

**ENCUESTA DE ESTUDIANTES[[1]](#footnote-1)**

Se tomó una muestra de los 300 estudiantes del 25%, para el desarrollo de la encuesta.

Se indago, ¿cuál es el tipo de solicitud más concurrente?

|  |  |  |
| --- | --- | --- |
| Motivo de la solicitud | Cantidad | % |
| **Atención Psicológica** | **3** | **3.9%** |
| **Administrativa** | **24** | **31.6%** |
| **Reparaciones** | **14** | **18.4%** |
| **Todas las anteriores** | **35** | **46.1%** |
| **Total General** | **76** | **100%** |

Tabla 3 Resultados Pregunta 1

¿Recibe usted notificación de la solución de estas solicitudes?, los resultados fueron:

|  |  |  |
| --- | --- | --- |
| Respuesta | Cantidad | % |
| **SI** | **2** | **2.6%** |
| **No** | **74** | **97.4%** |
| **Total General** | **76** | **100%** |

Tabla 4 Resultados pregunta 2

Acerca del tiempo de solución de estas solicitudes, los resultados son

|  |  |  |
| --- | --- | --- |
| Tiempo de Solución | Cantidad | % |
| **1 a 3 Días** | **14** | **18.4%** |
| **Más de 3 días** | **22** | **28.9%** |
| **1 Semana** | **37** | **48.7%** |
| **1 Mes** | **2** | **2.6%** |
| **Más de 1 Mes** | **1** | **1.3%** |
| **Total General** | **76** | **100%** |

Tabla 5 Resultados Pregunta 3

Ahora bien, con el ánimo de saber que inventario de elementos fueron asignados a cada espacio se les pregunto: ¿Sabe usted que elementos fueron asignados a su apartamento?, los resultados son:

|  |  |  |
| --- | --- | --- |
| Respuesta | Cantidad | % |
| **SI** | **21** | **27.6%** |
| **No** | **55** | **72.4%** |
| **Total General** | **76** | **100%** |

Tabla 6 Resultados Pregunta 4

Por otra parte, con el ánimo de saber sobre el proceso de reparaciones, se les pregunto:

¿Verifica las reparaciones realizadas en el apartamento?, los resultados son

|  |  |  |
| --- | --- | --- |
| Respuesta | Cantidad | % |
| **SI** | **35** | **46.1%** |
| **No** | **41** | **53.9%** |
| **Total General** | **76** | **100%** |

Tabla 7 Resultados de Pregunta 5

Como se aprecia el 77% de los encuestados manifiesta que el tiempo empleado para la solución de sus solicitudes es muy elevado, sin observar que existen casos aislados donde el tiempo es superior a un mes, para la atención de las solicitudes no existe la comunicación adecuada entre los administrativos y los estudiantes, por lo tanto, los procesos administrativos son deficientes, entre otras razones debido a la falta de tecnificación de los registros administrativos.

* 1. **Análisis de Documentos**

La CRU allegó documentación sobre los flujos de información manuales que

históricamente venían funcionando.

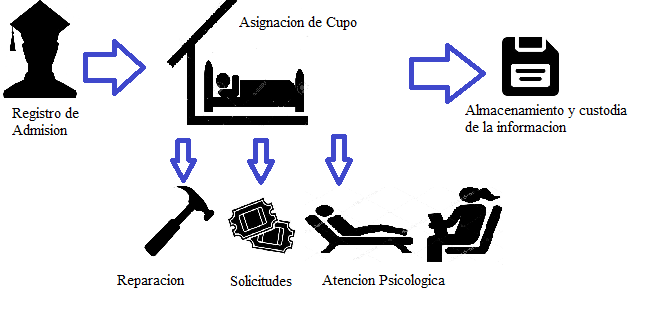


Ilustración 2 Flujo de información de la CRU

Donde se evidencio que solo usan un formato de admisión y con base en este, se da la creación de la carpeta de estudiante, donde se adjuntan los documentos mencionados en el proceso de admisión.

Para los demás procesos no se lleva ningún tipo de registro o evidencia de la ejecución de los mismos.

1. **INVESTIGACIÓN DE DEFINICIÓN DE TIPO PRODUCTO**

Al realizar estudios de mercado en cuatro (4) empresas proveedores de tecnología para

este tipo de sector económico, se encontró que existen algunos desarrollos sobre este

tema en particular, así:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Módulos** | **Ventajas** | **Desventajas** |
| **Sistema de Gestión de Espacios Físicos**  En la universidad pontificia universidad católica de chile, se encuentra el sistema de gestión de espacios físicos, donde mediante una url publica, se puede realizar una reserva de algún salón especial (conferencia, reunión), estos se encuentra en los diferentes bloques (PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE, 2015).  **TIPO: Sistema de gestión** | -Cuenta con un módulo de inicio de sesión, para restringir el ingreso de personas externas a la comunidad estudiantil, sin embargo, cuenta con un formulario de inscripción para personas que cuenten con autorización para realizar reservas o aun no estén inscritas.  - En el módulo principal, se realiza la reserva mediante un formulario básico donde se indica el espacio, la fecha de inicio y fin.  - Además de esto, también cuenta con un módulo de ayuda donde se presentan las indicaciones del formulario, y dudas que se tengan sobre la reserva. | -Tiene un sistema de seguridad, el cual cuenta con la conexión de un directorio activo donde están alojadas las direcciones de correo autorizadas para el ingreso a las reservas, este permite dar de baja los alumnos cada semestre | -Es poca la información acerca del espacio a reservar, puesto que no se tiene presente información de ubicación, capacidad de personas, etc.  -No se tiene un inventario puntual, de los elementos que se encuentran alojados en el espacio físico. |
| **Historiasclinicas.com**  Es un software web que gestiona de manera completa las historias clínicas, por medios formularios, donde se describen tratamientos, medicamentos, y demás fichas de médicas que interfieran con el análisis médico. (OPERALABORIS, 2016).  **TIPO:** Sistema de gestión de información | -Panel de administración: desde este módulo se logra una administración completa en cuanto a la agregación de los elementos que permiten gestionar la historia clínica, como lo son un médico, una categoría, o un grupo nuevo de medicamentos.  - Módulo de Pacientes, en este, se logra identidad toda la información del paciente donde por medio de grupos de fichas, se almacena de manera correcta los datos que involucren el diagnostico.  - Módulo de asignación de citas: por medio de este módulo se logra realizar la reserva de un tiempo de cita para cada doctor de acuerdo a la agenda disponible, para cada paciente. Además permite la consulta por día sobre cada paciente | -Permite almacenamiento online, donde se generan copias de back, de esta manera se garantiza la seguridad de la información.  - Permite la gestión de menús de acuerdo al perfil en el momento de la creación de usuarios. | - Los formularios son muy extensos, de esta manera se pierde la facilidad presentada hacia el usuario, donde se genera una sensación molestia al tener que diligenciar tantos campos  - No tiene una barra marcada acerca de la navegación actual, de esta manera el usuario puede llegar a olvidar en que formulario se encuentra. |
| **Helpdesk**  Este software tiene como función principal la gestión de las solicitudes, de acuerdo a las categorías, donde se logre realizar un seguimiento completo hasta su solución, donde por medio de informes se logren generar soluciones globales y así minimizar las solicitudes (ARTOLOGIK, 2016).  **TIPO:** Sistema de gestión de solicitudes | -Gestión de solicitudes: a través de este módulo se gestiona las solicitudes, por medio de formularios donde se indica las anotaciones, y sus respectivas soluciones.  -Base de conocimiento: donde se permite crear una lista de soluciones básicas y a partir de solicitudes cerradas, se permite buscar de inmediato la solución y así mitigar los problemas que se generen.  -Informes: en este módulo se encuentra la posibilidad de generación de reportes, donde se indiquen, cantidad de solicitudes, tiempos de solución, además de la categoría más solicitada. | -Tiene opciones de escalamiento con el fin de la solución entre varias áreas, sin perder el control de que quien tiene la solución.  -Tiene funcionalidad web, lo cual garantiza una conexión a través de un explorador de internet. | -No permite la integración con más módulos, puesto que de acuerdo las solicitudes se pueden presentar otros como lo es de inventarios  -En el módulo de informes no tiene la opción de crear informes personalizados |
| **Gestockplus**  Es un software de gestión comercial el cual tiene como principal función administrar el inventario de elementos, a través de diferentes módulos (GESTOCK PLUS, 2016).  **TIPO:** Sistema de control de inventarios | -Gestión de Existencias: Por medio de este módulo se permite la verificación de elementos disponibles, faltantes, y así mismo visualizar un listado de existencias, con sus respectivas características.  -Gestión de Clientes: este módulo permite la clasificación de grupos de clientes, para así lograr determinar el inventario por cada cliente y estimar el próximo envió, y así mismo asociarlos mediante una ficha o descripción de cada uno.  -Estadísticas en Informes: Permite generar reportes por cliente o por elementos, para así determinar cuál es el más vendido. También permite verificar movimientos mensuales o personalizar estos tiempos. | -El tipo de software el local, lo cual significa que la información siempre permanecerá en la red de la entidad que lo administre.  - Tiene una interfaz amigable con el usuario, además de sus accesos directos con el teclado, lo cual agiliza los procesos. | -Debido a que es un software local, no permite la compatibilidad con demás sistemas operativos o arquitectura. Esta versión solo está para sistema operativo Windows 7 de 32 bits  -Difícil de administrar si se realizan actualizaciones, puesto que se necesitaría reinstalar el software de manera individual en cada uno de los equipos. |
| **Fronthotel**  Es un software diseñado para la administración de hoteles, se caracteriza por su gran sistematización de información de todas las áreas que posea. Además de esto cuenta con motor de bases de datos SQL Server (ICG Software, 2016).  **TIPO:** Sistema de gestión de hospedaje | **-Recepción:** Este módulo permite el ingreso de los huéspedes, a través de la consulta inmediata de la disponibilidad, tamaños y precios de las habitaciones.  **-Tarifas:** Este módulo permite gestionar los valores que tendrán las habitaciones de acuerdo a las temporadas, además permite generar combos con el fin de ir al detalle de la ganancia.  **-Gobernanta:** Por medio de este módulo, se puede verificar el estado de las habitaciones, además de los objetos incluidos en cada una. | -Permite administración desde cualquier punto del mundo gracias a que es un aplicativo web.  -Permite integración con cualquier tipo de dispositivo de receptor de información (Scanner, datafono, pistola lectora). | -Debido a la fácil integración con todas las áreas del hotel, requiere un soporte técnico dedicado, lo cual representa costos adicionales.  -Gracias a su gran variedad de módulos para las áreas del hotel, pierde el objetivo principal, por tanto, no tiene un límite de operación. |

Ante este panorama, la compañía decidió contratar una solución ligera, tanto en el aspecto económico como de infraestructura, pero que permita la recolección, centralización y procesamiento de la información crucial para la CRU.

Realizando un análisis del sector, sobre la oferta y demanda de servicios que requiere el

proyecto, se comparó una solución con infraestructura en sitio, frente a una solución del tipo computación en la nube con el fin de encontrar aquella que presentará una mejor condición técnica, administrativa y financiera.

En esta comparación se tuvieron en cuenta los costos directos e indirectos que cada una

de las soluciones requiere para su implementación, tales como el valor del hardware, software, energía eléctrica, conectividad, administración, entre otros.

La delimitación económica se establece mediante la disponibilidad de los recursos a destinar para el desarrollo de todas las actividades que involucre el cumplimiento del cronograma:

|  |  |  |
| --- | --- | --- |
| Recurso | Cantidad | Costo |
| Director de Proyecto  JEFERSSON GUEVARA | 720 Horas / Vlr Hr 20.000 | 14.400.000 |
| Analista funcional  JEFERSSON GUEVARA | 320 Horas / Vlr Hr 40.000 | 12.800.000 |
| Desarrollador  JEFERSSON GUEVARA | 320 Horas / Vlr Hr 50.000 | 16.000.000 |
| Tester  JEFERSSON GUEVARA | 320 Horas / Vlr Hr 20.000 | 6.400.000 |
| Equipo | 1 | 2.000.000 |
| Microsoft Office | 6 Meses | 180.00 |
| Licencia de Visual Studio  Commnity 2017 | 6 Meses | 0.00 |
| Windows 10 Pro | 6 Meses | 720.000 |
| SQL Server Developer 2017 | 6 Meses | 0.00 |
| Servicio de Internet | 6 Meses | 480.000 |
| Luz | 6 Meses | 900.000 |
| Agua y Alcantarillado | 6 Meses | 360.000 |
| Papelería y otros | 1 | 300.000 |
| Instalaciones | 6 Meses Arriendo | 1.800.000 |
| Transporte | 1 | 500.000 |
| Total |  | 58.140.000 |

Tabla 8 Delimitación Económica

Como se observa las licencias de los entornos de desarrollo tiene un costo de $0.00, esto debido a que en la ejecución del proyecto solo estará trabajando un desarrollador, y por consecuencia no supera los cinco integrantes del grupo que permite Microsoft para el uso de estas licencias.

Además de los costos de ejecución del proyecto se prevén gastos anuales de infraestructura y mantenimiento, donde existen dos opciones, una en la Nube y otra en la instalación en sitio en la CRU, con el fin de identificar los costos más bajos se identifican los siguientes elementos:

|  |  |  |
| --- | --- | --- |
| Recurso | Cantidad | Costo |
| Hosting, Api, y Sitio 10GB  1 Core, 1.75 Ram  480 Horas | Mes | 225.000 |
| Base de datos única, Estándar tier, S0 level, 10 DTUs, 250 GB | Mes | 30.000 |
| Almacenamiento 20GB | Mes | 5.000 |
| Ancho de Banda | Mes | 0.00 |
| Administración | 5 Días por Mes | 400.000 |
|  | Total Mensual | 660.000 |
|  | Total Anual | 7’920.000 |

Tabla 9 Costos de Plataforma Azure

|  |  |  |
| --- | --- | --- |
| Recurso | Cantidad | Costo |
| Servidor  PowerEdge-T130 | 1 | 6’200.000 |
| Licencia Windows  Server2016 | 1 | 2’700.000 |
| Licencia de Sql Server 2014 | 1 | 2’880.000 |
| Energía | Mensual | 90.000 |
| Internet 10MB | Mensual | 67.000 |
| Administración | Mensual | 1.800.000 |
| Instalación de  Infraestructura | 1 | 2’000.000 |
|  | Total Mensual | 1.957.000 |
|  | Total costo único | 13’780.000 |
|  | Total Anual | 23’484.000 |
|  | Valor Total | 37’264.000 |

Tabla 10 Costos Implementación en sitio

Como se observa, los costos de implementación en la nube en la plataforma de Azure, son más bajos que la implementación en sitio, puesto que la primera es un 21% del costo anual de la solución en sitio, y es más accesible económicamente.

1. **REQUERIMIENTOS**
   1. **Requerimientos funcionales**

|  |  |
| --- | --- |
| Nombre | RF01-Gestion de los elementos |
| Resumen | Se debe permitir agregar, consultar y actualizar elementos por espacio, donde se mencione la descripción y cantidad y la fecha en que se le asigno. |
| Entradas | |
| Descripción, Cantidad, Nombre de Espacio | |
| Resultados | |
| Reportes Elementos por espacio y por elemento | |
|  |  |
| Nombre | RF02-Gestion de Expedientes |
| Resumen | Se debe realizar la actualización y consulta de la información personal, acudientes, universidad y solicitudes o citas asociadas durante el proceso de estancia en la residencia. |
| Entradas | |
| Número de identificación del estudiante | |
| Resultados | |
| Reporte del expediente para consulta | |
|  |  |
| Nombre | RF03-Gestion de solicitudes administrativas |
| Resumen | Se debe permitir el registro, consulta y actualización de las solicitudes administrativas que genere los estudiantes, también realizar el registro anotaciones a la solicitud con el fin de tener un histórico de la misma y dar solución |
| Entradas | |
| Descripción de la solicitud, datos adicionales. | |
| Resultados | |
| Numero de solicitud, con tiempo máximo de solución y responsable de la solución.  Se agrega un registro con la anotación, la fecha y hora, el empleado que realiza la anotación. | |
|  |  |
| Nombre | RF04-Gestion de Citas Psicológicas |
| Resumen | Se debe permitir el agendamiento, consulta y actualización de las citas psicológicas genere los estudiantes  Debe permitir realizar el registro anotaciones de tipo expediente, con el fin de documentar toda la cita |
| Entradas | |
| Fecha, Psicólogo, descripción breve del motivo, datos adicionales.  Anotación y solución | |
| Resultados | |
| Numero de cita, Fecha, Psicólogo | |
|  |  |
| Nombre | RF05-Gestion de reparaciones |
| Resumen | Se debe permitir el registro, consulta y actualización de las reparaciones que genere los estudiantes en los espacios asignados.  Debe permitir realizar el registro anotaciones a la solicitud de reparación , con el fin de documentar todo lo que se realizó en la reparación incluyendo el costo total de los elementos comprados de ser necesario |
| Entradas | |
| Descripción del daño presentado, fecha y hora de disponibilidad.  Anotación, costo, solución. | |
| Resultados | |
| Numero de solicitud de reparación, tiempo máximo de solución y nombre de responsable.  Se agrega un registro con la anotación, la fecha y hora, el empleado que realiza la anotación y la solicitud cambia de estado | |

|  |  |
| --- | --- |
| Nombre | RF06-Gestion de Admisiones |
| Resumen | Debe permitir realizar el registro del formulario de admisión público, de manera que cualquier estudiante interesado lo diligencie sin necesidad usuario y contraseña.  También actualizar la gestión realizada por los administrativos para emitir el resultado de la admisión. |
| Entradas | |
| Formulario diligenciado | |
| Resultados | |
| Admisión registrada y lista para evaluar, resultado de la admisión | |

* 1. **Requerimientos no funcionales**
     1. **Motor de Base de Datos**

Con el fin de garantizar niveles de transacción, configuración, administración y soporte técnico del fabricante este proyecto se realizará en Microsoft SQL Server 2016 Developer Edition, ya que es un sistema de administración de datos, eficaz y confiable, que ofrece un almacén de datos completo para sitios web ligeros, siendo idóneo para desarrollar y activar pequeñas aplicaciones de servidor, web y de escritorio.

* + 1. **Software de Desarrollo**

Con el fin de logar compatibilidad nativa con el motor de base de datos, así como confiabilidad y soporte técnico del fabricante este proyecto se programará en el entorno de desarrollo de Microsoft Visual Studio Community 2017, ya que permite crear aplicaciones atractivas del gusto de los usuarios.

* + 1. **Infraestructura**

Con el fin de permitir el acceso adaptado y bajo demanda a un conjunto compartido de recursos de computación configurables compartidos este proyecto estará bajo el modelo tecnológico: Infraestructura como servicio, sobre la plataforma AZURE[[2]](#footnote-2) a través de un proveedor de servicio de internet.

* + 1. **Seguridad**

Por Seguridad se hace referencia a la habilidad del sistema de controlar el acceso a los servicios e información del negocio; así como la capacidad de detectar, aislar y restablecer continuidad ante una falla de seguridad.

La aplicación debe cumplir con unos requisitos mínimos de seguridad Estas características se describen a continuación:

* Seguridad de acceso al sistema

De acuerdo al nivel de seguridad, la aplicación permitirá a los usuarios registrados en el Sistema de acuerdo al ROL el ingreso hacia las diversas funcionalidades.

o Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador de acceso a datos.

* Verificación de campos obligatorios:

El sistema debe verificar que los datos correspondientes a los campos obligatorios de los formularios que fueron ingresados por el usuario final.

* Verificación de campos numéricos y de texto.

El sistema debe verificar que los campos correspondientes a datos exclusivamente numéricos no contengan caracteres de texto o caracteres especiales tales como comas, puntos, asteriscos.

* Navegación en el sistema

El usuario solo podrá navegar entre las páginas del sistema a través de las opciones que le presenta la aplicación y solo podrá acceder a aquellas autorizadas para el rol correspondiente.

* Manejo de Contraseñas

El sistema debe proveer las funcionalidades de autenticación. Para tal fin debe proveer las interfaces de usuario necesarias para permitir las siguientes funcionalidades:

- Pantalla para autenticación de usuarios en el sistema (usuario y contraseña)

- Pantalla para cambio de contraseña del usuario. Debe pedir la contraseña anterior y la nueva contraseña.

- Retroalimentación hacia el usuario si se producen errores durante el proceso de autenticación.

- El sistema debe validar los datos ingresados en la url, para evitar inyección de código sql y js.

* + 1. **Diseño Corporativo**

La solución web y la aplicación, debe contener colores corporativos para los títulos, fondos, botones, y demás formularios que se implementen, de acuerdo a los ya establecidos por la CRU.

* + 1. **Disponibilidad**

Con el fin de disminuir los costos de uso de la plataforma en la nube se prevé el uso de la plataforma en un horario 7 días a la semana 24 horas del día, durante este tiempo la disponibilidad será del 98%.

• El tiempo para iniciar o reiniciar el sistema no podrá ser mayor a 5 minutos.

• La tasa de tiempos de falla del sistema no podrá ser mayor al 0,5% del tiempo de operación total.

• El promedio de duración de fallas no podrá ser mayor a 15 minutos.

• La probabilidad de falla del Sistema no podrá ser mayor a 0,05.

* + 1. **Usabilidad**

El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 4 horas.

La tasa de errores cometidos por el usuario deberá ser menor del 1% de las transacciones totales ejecutadas en el sistema.

El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final.

La aplicación web debe poseer un diseño “Responsive” a fin de garantizar la adecuada visualización en múltiples computadores personales, dispositivos tableta y teléfonos inteligentes.

El sistema debe poseer interfaces gráficas bien formadas.

* + 1. **Eficiencia**

• Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos.

• Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 2 segundos.

# **MODELAMIENTO DE BASE DE DATOS**

## **Modelo Relacional**

Ver diagrama

## **Script de Base de Datos**

create database PrototipoCRU

go

use PrototipoCRU

go

create table Permiso

( Permiso\_id Int Identity(1,1) not null primary key,

Modulo varchar (50) not null,

descripcion varchar(50) not null

);

go

insert into Permiso

(Modulo, descripcion)

values

('Empleado', 'Puede Crear'),

('Empleado', 'Puede Consultar'),

('Empleado', 'Puede Actualizar'),

('Espacio', 'Puede Crear'),

('Espacio', 'Puede Consultar'),

('Espacio', 'Puede Actualizar'),

('Estudiante', 'Puede Crear'),

('Estudiante', 'Puede Consultar'),

('Estudiante', 'Puede Actualizar'),

('Solicitud', 'Puede Crear'),

('Solicitud', 'Puede Consultar'),

('Solicitud', 'Puede Actualizar'),

('Admision', 'Puede Crear'),

('Admision', 'Puede Consultar'),

('Admision', 'Puede Actualizar'),

('Citas', 'Puede Crear'),

('Citas', 'Puede Consultar'),

('Citas', 'Puede Actualizar'),

('Reparacion', 'Puede Crear'),

('Reparacion', 'Puede Consultar'),

('Reparacion', 'Puede Actualizar'),

('Inventario', 'Puede Crear'),

('Inventario', 'Puede Consultar'),

('Inventario', 'Puede Actualizar'),

('Elemento', 'Puede Crear'),

('Elemento', 'Puede Consultar'),

('Elemento', 'Puede Actualizar')

go

create table tipoidentificacion

(

id\_tipoidentificacion int IDENTITY(1,1) not null,

valor\_tipoidentificacion varchar(50) not null,

primary key (id\_tipoidentificacion)

);

go

INSERT INTO tipoidentificacion

(valor\_tipoidentificacion)

VALUES

('Cedula'),

('Tarjeta de Identidad'),

('Cedula Extranjeria')

go

create table estadocivil

(

id\_estadocivil int Identity(1,1) not null,

valor\_estadocivil varchar(20) not null,

primary key (id\_estadocivil)

);

go

INSERT INTO estadocivil

(valor\_estadocivil)

VALUES

('Soltero'),

('Casado'),

('Union libre'),

('Divorciado'),

('Viudo')

go

CREATE TABLE estado\_usuario\_cru

(

Id\_estado\_usuario\_cru int Identity(1,1),

Descripcion\_estado\_usuario\_cru varchar(30) NOT NULL,

primary key (Id\_estado\_usuario\_cru)

);

GO

INSERT INTO estado\_usuario\_cru

(Descripcion\_estado\_usuario\_cru)

VALUES

('Activo'),

('Retirado'),

('Admitido'),

('En proceso Admision'),

('Expulsado'),

('Inactivo'),

('Inicio Sesion'),

('Cerro Sesion')

go

create table piso

(

id\_piso int IDENTITY(1,1) not null,

valor\_piso varchar (2) not null,

primary key (id\_piso)

);

INSERT INTO piso

(valor\_piso)

VALUES

('1'), ('2'), ('3'), ('4'), ('5'), ('6'), ('7'), ('8'), ('9'), ('10'), ('11'), ('12'), ('13'

go

create table cargo

(

id\_cargo int IDENTITY(1,1) not null,

valor\_cargo varchar(20) not null,

primary key (id\_cargo)

);

INSERT INTO cargo

(valor\_cargo)

VALUES

('Psicologo'),

('Gestor Social'),

('Mantenimiento'),

('Director')

go

create table tipo\_expediente

(

id\_tipo\_exp int IDENTITY(1,1) not null,

valortipoexp varchar (50) not null,

primary key (id\_tipo\_exp)

);

INSERT INTO tipo\_expediente

(valortipoexp)

VALUES

('Psicologico'),

('Social')

go

create table rol

(

id\_rol int IDENTITY(1,1) not null,

descripcion\_rol varchar (50) not null,

primary key (id\_rol)

);

go

INSERT INTO rol

(descripcion\_rol)

VALUES

('Empleado'),

('Estudiante'),

('Mantenimiento'),

('Administrador')

go

create table Permiso\_Rol

(

Rol\_id int not null,

Permiso\_id\_ int not null,

primary key (Rol\_id, Permiso\_id\_),

CONSTRAINT fk\_rol\_Permiso

FOREIGN KEY ( Rol\_id )

REFERENCES rol ( id\_rol),

CONSTRAINT fk\_Permiso\_Rol

FOREIGN KEY ( Permiso\_id\_ )

REFERENCES Permiso ( Permiso\_id)

);

go

insert into Permiso\_Rol

(Rol\_id,Permiso\_id\_)

values

(1, 1), (1, 3), (1, 4), (1, 6), (1, 7)

go

create table estado\_solicitud

(

id\_estado\_solciitud int IDENTITY(1,1) not null,

valor\_estado\_solciitud varchar (50) not null,

primary key (id\_estado\_solciitud)

);

INSERT INTO estado\_solicitud

(valor\_estado\_solciitud)

VALUES

('Recibido'),

('Atendiendo'),

('Suspendido'),

('Solucionado'),

('Cancelado'),

('Proceso de aprobacion'),

('vencido')

go

create table prioridad\_solicitud

(

id\_prioridad\_solciitud int IDENTITY(1,1) not null,

valor\_prioridad\_solciitud varchar (50) not null,

tiempodesolucion\_horas int not null,

primary key (id\_prioridad\_solciitud)

);

go

INSERT INTO prioridad\_solicitud

(valor\_prioridad\_solciitud, tiempodesolucion\_horas)

VALUES

('Alta', 24),

('Media', 48),

('Baja', 72),

('Urgente', 8)

go

create table tipo\_espacio

(

id\_tipo\_espacio int IDENTITY(1,1) not null,

valor\_tipo\_espacio varchar (20) not null,

primary key (id\_tipo\_espacio)

);

go

INSERT INTO tipo\_espacio

(valor\_tipo\_espacio)

VALUES

('Residencial'),

('Recreativo'),

('Almacenamiento')

go

create table tipo\_elemento

(

id\_tipo\_elemento int IDENTITY(1,1) not null,

valor\_tipo\_elemento varchar (30) not null,

primary key (id\_tipo\_elemento)

);

go

INSERT INTO tipo\_elemento

(valor\_tipo\_elemento)

VALUES

('Uso Residencial'),

('Uso Recreativo'),

('Tecnologico')

go

Create Table departamento

(

id\_departamento int IDENTITY(1,1) not null,

descripcion\_departamento varchar (100) not null,

primary key (id\_departamento)

);

go

INSERT INTO departamento

(descripcion\_departamento)

VALUES

('ANTIOQUIA'), ('ATLANTICO'),

('BOGOTA'), ('BOLIVAR'),

('BOYACA'), ('CALDAS'),

('CAQUETA'), ('CAUCA'), ('CESAR'), ('CORDOBA'), ('CUNDINAMARCA'), 'CHOCO'), ('HUILA'), ('GUAVIARE'), ('MAGDALENA'), ('META'), ('NARIÑO'),

('N. DE SANTANDER'), ('QUINDIO'), ('RISARALDA'), ('SANTANDER'), ('SUCRE'), ('TOLIMA'), ('VALLE DEL CAUCA'), ('ARAUCA'), ('CASANARE'), ('PUTUMAYO'), ('SAN ANDRES'), ('AMAZONAS'), ('GUAINIA'), ('LA GUAJIRA'), ('VAUPES'), ('VICHADA')

go

create table raza

(

id\_raza int IDENTITY (1,1) not null,

descripcion\_raza varchar (14) not null,

primary key (id\_raza)

);

go

INSERT INTO raza

(descripcion\_raza)

VALUES

('Afrocolombiano'), ('Raizal'), ('Indigena'),

('Mulato'),

('Rom'),

('Negro'),

('Ninguno')

go

create table tipodesangre

(

id\_tipo\_sangre int IDENTITY(1,1) not null,

descripcion\_tipo\_sangre varchar(11) not null,

primary key(id\_tipo\_sangre)

);

go

INSERT INTO tipodesangre

(descripcion\_tipo\_sangre)

VALUES

('O -'), ('O +'), ('A -'), ('A +'), ('B -'), ('B +'), ('AB -'), ('AB +')

go

create table tipodevivienda

(

id\_tipodevivienda int IDENTITY(1,1) not null,

descripcion\_tipovivienda varchar(9) not null,

primary key (id\_tipodevivienda)

);

INSERT INTO tipodevivienda

(descripcion\_tipovivienda)

VALUES

('Arrendada'), ('Familiar'), ('Propia')

go

create table estado\_espacio

(

id\_estado\_espacio int IDENTITY (1,1) not null primary key,

valor\_estado\_espacio varchar (10) not null,

);

go

insert into estado\_espacio

(valor\_estado\_espacio)

values

('Disponible'), ('Ocupado')

go

create table espacio

( id\_espacio int IDENTITY(1,1) not null,

descripcion\_espacio varchar (100) not null,

capacidad int ,

cupo int ,

id\_tipo\_espacio\_ int not null,

id\_piso\_espacio int not null,

primary key (id\_espacio),

CONSTRAINT fk\_espacio\_piso

FOREIGN KEY ( id\_piso\_espacio )

REFERENCES piso ( id\_piso),

CONSTRAINT fk\_tipo\_espacio

FOREIGN KEY ( id\_tipo\_espacio\_ )

REFERENCES tipo\_espacio ( id\_tipo\_espacio)

);

go

create table Persona

(

id\_persona int Identity(1,1)not null,

correo\_directorio varchar(300) not null,

contrasena varchar(50) not null,

id\_rol\_Directorio int not null,

NumeroIdentificacion varchar(15) not null,

Nombres varchar (100) not null,

Apellidos varchar (100) not null,

Estadocivil int not null,

--Tabla EstadoCivil

Estrato int not null,

Direccion varchar (100) not null,

Telefono varchar (20) not null,

Tipo\_sangre int not null,

Fechanacimiento datetime not null,

MunicipioNacimiento varchar(100) not null,

DepartamentoNacimiento int not null,

Paisnacimiento int not null,

sesion int not null,

--Tabla Pais

primary key( id\_persona),

CONSTRAINT fk\_directorio\_Rol

FOREIGN KEY ( id\_rol\_Directorio )

REFERENCES rol ( id\_rol),

CONSTRAINT fk\_estadocivil\_Directorio

FOREIGN KEY ( Estadocivil )

REFERENCES estadocivil ( id\_estadocivil),

CONSTRAINT fk\_tiposangre\_Directorio

FOREIGN KEY ( Tipo\_sangre)

REFERENCES tipodesangre ( id\_tipo\_sangre ),

CONSTRAINT fk\_pais\_Directorio

FOREIGN KEY ( Paisnacimiento)

REFERENCES Pais ( id\_pais ),

CONSTRAINT fk\_departamento\_Directorio

FOREIGN KEY ( departamentoNacimiento)

REFERENCES departamento ( id\_departamento )

);

go

create table HistoricoTipoIdentificacion(

id int IDENTITY(1,1) not null,

id\_personatipo int not null,

id\_tipoidentifica int not null,

primary key (id),

CONSTRAINT fk\_tipodocumentoDirectorio

FOREIGN KEY ( id\_tipoidentifica )

REFERENCES tipoidentificacion ( id\_tipoidentificacion ),

CONSTRAINT fk\_hisotiropersonatipoiden

FOREIGN KEY ( id\_personatipo )

REFERENCES Persona ( id\_persona )

);

go

create table empleado

(

id\_empleado int IDENTITY(1,1) not null,

empleado\_directorio int not null,

--Tabla Directorio

primary key (id\_empleado),

CONSTRAINT fk\_Empleado\_directorio

FOREIGN KEY (empleado\_directorio )

REFERENCES Persona ( id\_persona )

);

go

create table contrato

(

id\_contrato int IDENTITY(1,1) NOT null,

id\_empleado int not null,

cargo\_contrato int not null,

-- Tabla Cargo

fecha\_inicio date not null,

fecha\_fin date not null,

tipo\_de\_contrato varchar(50) not null,

salario bigint,

PRIMARY key (id\_contrato),

CONSTRAINT fk\_cargo\_contrato

FOREIGN KEY (cargo\_contrato )

REFERENCES cargo ( id\_cargo ),

CONSTRAINT fk\_empelado\_contrato

FOREIGN KEY (id\_empleado )

REFERENCES empleado ( id\_empleado )

);

GO

create table historico\_directorio(

id\_historico\_directorio int IDENTITY(1,1) not null,

id\_directorio\_ int not null,

Estado\_directorio\_CRU int not null,

descripcion\_histoico varchar (200)not null,

fecha date not null,

id\_empleado int,

-- Estado

primary KEY (id\_historico\_directorio),

CONSTRAINT fk\_directorio\_historico

FOREIGN KEY ( id\_directorio\_ )

REFERENCES Persona ( id\_persona),

CONSTRAINT fk\_directorio\_estado

FOREIGN KEY ( Estado\_directorio\_CRU )

REFERENCES estado\_usuario\_cru ( Id\_estado\_usuario\_cru),

CONSTRAINT fk\_historico\_mod\_empleado

FOREIGN KEY ( id\_empleado )

REFERENCES empleado ( id\_empleado)

);

create table estudiante

(

id\_estudiante int IDENTITY(1,1) not null,

Servicio\_Salud varchar (200) not null,

Dispacidad\_estudiante varchar (2) not null,

descripcion\_dispacacidad\_estudainte varchar (200),

Situaciondesplazamientoestudiante varchar (2) not null,

Numerohermanos int not null,

tipodevivienda\_estudiante int not null,

--Tabla Tipovivienda

apoyouniversidad varchar (2) not null,

descripcion\_apoyo\_ varchar(100),

raza\_estudiante int not null,

id\_directorio\_estudiante int not null,

--Tabla

primary key (id\_estudiante),

CONSTRAINT fk\_estudiante\_directorio

FOREIGN KEY (id\_directorio\_estudiante )

REFERENCES Persona ( id\_persona ),

CONSTRAINT fk\_tipovivienda\_estudiante

FOREIGN KEY ( tipodevivienda\_estudiante )

REFERENCES tipodevivienda ( id\_tipodevivienda ),

CONSTRAINT fk\_raza\_estudiante

FOREIGN KEY ( raza\_estudiante )

REFERENCES raza ( id\_raza)

);

go

create table historicoestudiante

(

id\_historico\_expediente int IDENTITY(1,1) not null,

id\_estudiante int not null,

fecha\_historico\_expediente datetime not null,

descripcion\_historico\_expediente varchar (500) not null,

id\_empleado\_historicoestudiante int,

primary key (id\_historico\_expediente),

CONSTRAINT fk\_estudiante\_historico

FOREIGN KEY (id\_estudiante )

REFERENCES estudiante ( id\_estudiante ),

CONSTRAINT fk\_empleado\_historicoestudiante

FOREIGN KEY (id\_empleado\_historicoestudiante )

REFERENCES empleado( id\_empleado )

);

CREATE TABLE historico\_espacio

(

id\_historico\_espacio int IDENTITY(1,1) not null,

fecha\_historico\_espacio date not null,

estado\_espacio\_o int not null,

descripcion varchar (100) not null,

id\_espacio\_historico int not null,

id\_empleado\_historico int,

id\_estudiante\_espacio int,

PRIMARY KEY (id\_historico\_espacio),

CONSTRAINT fk\_estado\_espacio

FOREIGN KEY ( estado\_espacio\_o )

REFERENCES estado\_espacio ( id\_estado\_espacio),

CONSTRAINT fk\_empleado\_historico

FOREIGN KEY ( id\_empleado\_historico )

REFERENCES empleado ( id\_empleado),

CONSTRAINT fk\_historico\_espacio

FOREIGN KEY ( id\_espacio\_historico )

REFERENCES espacio ( id\_espacio),

CONSTRAINT fk\_estudiante\_espacio

FOREIGN KEY ( id\_estudiante\_espacio )

REFERENCES estudiante ( id\_estudiante)

);

GO

create table datosuniversidad

(

id\_datosuniversidad int IDENTITY (1,1) not null primary key,

Universidad\_estudiante varchar (200) not null,

Facultad\_estudiante varchar (200) not null,

Programa\_estudiante varchar (200) not null,

Puntaje\_Basico\_Matricula varchar (20) not null,-- 0 a 50

Promedio\_Academico\_estudiante varchar (20),

Año\_ingreso\_Universidad\_ datetime not null,

semestre\_ingreso\_universidad int not null,

Porcentaje\_Avance varchar (2),

id\_estudiante\_datosuniversidad int not null ,

CONSTRAINT fk\_datosuniversidad\_estudiante

FOREIGN KEY ( id\_estudiante\_datosuniversidad )

REFERENCES estudiante ( id\_estudiante)

);

go

create table acudiente

(

id\_acudiente int IDENTITY(1,1) not null,

dependencia\_econo\_ varchar (2) not null,

nombre\_acudiente varchar (100) not null,

apellidos\_acudiente varchar (100) not null,

ocupacion\_acudiente varchar (100) not null,

direccion\_acudiente varchar(200) not null,

departamento\_acudiente int not null,

telefono\_acudiente varchar (20) not null,

parentezo\_acudiente varchar (100) not null,

correo\_acudiente varchar (100) not null,

primary key (id\_acudiente),

CONSTRAINT fk\_acudiente\_departamento

FOREIGN KEY ( departamento\_acudiente )

REFERENCES departamento ( id\_departamento)

);

go

create table acudiente\_estudainte(

id\_acudiente\_estudainte int IDENTITY(1,1) not null,

id\_estudiante int not null,

id\_acudiente int not null,

primary key (id\_acudiente\_estudainte),

CONSTRAINT fk\_estudiante\_acudiente

FOREIGN KEY ( id\_acudiente )

REFERENCES acudiente ( id\_acudiente),

CONSTRAINT fk\_acudiente\_estudiante

FOREIGN KEY ( id\_estudiante )

REFERENCES estudiante ( id\_estudiante)

);

Go

create table archivo\_estudiante

(

id\_archivo int not null,

id\_estudiante\_archi int not null,

descripcion\_archivo varchar (100) not null,

direccion\_archivo varchar (500) not null,

primary key (id\_archivo),

CONSTRAINT fk\_archivo\_estudiante

FOREIGN KEY ( id\_estudiante\_archi )

REFERENCES estudiante ( id\_estudiante ),

);

go

create table elemento

(

id\_elemento int IDENTITY(1,1) not null,

id\_tipo\_elemento\_ int not null,

descripcion varchar(50) not null,

marca varchar(50) ,

Modelo varchar (50),

primary key (id\_elemento),

CONSTRAINT fk\_tipo\_elemento

FOREIGN KEY ( id\_tipo\_elemento\_ )

REFERENCES tipo\_elemento ( id\_tipo\_elemento)

);

go

create table inventario\_espacio

(

id\_inventario\_esp int IDENTITY(1,1) not null,

id\_elemento\_inventario\_espacio int not null,

cantidad\_inventario\_elemento\_espacio int not null,

fecha\_inventario\_espacio datetime not null,

id\_espacio\_inventario\_\_ int not null,

primary key (id\_inventario\_esp),

CONSTRAINT fk\_elemento\_inventario

FOREIGN KEY ( id\_elemento\_inventario\_espacio )

REFERENCES elemento ( id\_elemento),

CONSTRAINT fk\_espacio\_inventario

FOREIGN KEY ( id\_espacio\_inventario\_\_)

REFERENCES espacio ( id\_espacio)

);

go

create table historio\_de\_inventario

(

id\_inventario\_historico int Identity(1,1) primary key,

id\_empleado\_historico\_inventario int not null,--

descripcion\_historico\_inventario varchar (200) not null,

fecha\_historico\_inventario datetime not null,

inventario\_id\_ int not null,--

CONSTRAINT fk\_historico\_empleado

FOREIGN KEY ( id\_empleado\_historico\_inventario )

REFERENCES empleado (id\_empleado),

CONSTRAINT fk\_historico\_inventario

FOREIGN KEY ( inventario\_id\_)

REFERENCES inventario\_espacio (id\_inventario\_esp)

);

go

create table solicitud

(

id\_solicitud int IDENTITY(1,1) not null,

id\_prioridad\_solciitud\_\_ int not null,

fecha\_solicitud date not null,

descripcion\_solicitud varchar (200) not null,

id\_estudiante\_solicitud int not null,

id\_empleado\_solicitud int not null,

primary key (id\_solicitud),

CONSTRAINT fk\_prioridad\_solicitud

FOREIGN KEY ( id\_prioridad\_solciitud\_\_ )

REFERENCES prioridad\_solicitud (id\_prioridad\_solciitud ),

CONSTRAINT fk\_empleado\_solicitud

FOREIGN KEY ( id\_empleado\_solicitud)

REFERENCES empleado ( id\_empleado),

CONSTRAINT fk\_estudiante\_solcitud

FOREIGN KEY ( id\_estudiante\_solicitud )

REFERENCES estudiante (id\_estudiante )

);

go

create table reparacion\_espacio

(

id\_reparacion int IDENTITY(1,1) not null,

fecha\_reparacion\_espacio datetime not null,

descripcion\_reparacion varchar (500) not null,

id\_solicitud\_reparacion int not null,

espacio\_reparacion int not null,

archivo\_Reparacion varchar(500),

antes varchar(100),

despues varchar (100),

costo bigint not null,

primary key (id\_reparacion),

CONSTRAINT fk\_espacio\_reparacion

FOREIGN KEY ( espacio\_reparacion)

REFERENCES espacio ( id\_espacio),

CONSTRAINT fk\_solicitud\_reparacion

FOREIGN KEY ( id\_solicitud\_reparacion)

REFERENCES solicitud( id\_solicitud)

);

go

create table historico\_solicitud

(

id\_historico\_solicitud int IDENTITY(1,1) not null,

id\_caso\_anotacion int not null,

descripcion\_anotacion varchar (200) not null,

fecha\_modificacion datetime not null,

id\_empleado\_historico int,

id\_estado\_solicitud\_ int,

primary key (id\_historico\_solicitud),

CONSTRAINT fk\_solcitud\_historio

FOREIGN KEY ( id\_caso\_anotacion)

REFERENCES solicitud ( id\_solicitud),

CONSTRAINT fk\_estado\_solicitud

FOREIGN KEY ( id\_estado\_solicitud\_ )

REFERENCES estado\_solicitud ( id\_estado\_solciitud),

CONSTRAINT fk\_empleado\_historio

FOREIGN KEY ( id\_empleado\_historico)

REFERENCES empleado ( id\_empleado)

);

go

create table solucion

(

id\_solucion int IDENTITY(1,1) not null,

descripcion\_solucion varchar(100) not null,

id\_empleado\_solucion int,

fecha\_solucion datetime,

primary key (id\_solucion),

id\_solicitud\_solucion int ,

CONSTRAINT fk\_solucion\_solicitud

FOREIGN KEY ( id\_solicitud\_solucion)

REFERENCES solicitud ( id\_solicitud),

CONSTRAINT fk\_empleado\_solucion

FOREIGN KEY ( id\_empleado\_solucion)

REFERENCES empleado ( id\_empleado),

);

go

create table estado\_cita

(

id\_estado\_cita int IDENTITY(1,1) not null,

valor\_estado\_cita varchar (50) not null,

primary key (id\_estado\_cita)

);

go

INSERT INTO estado\_cita

(valor\_estado\_cita)

VALUES

('Agendada'),

('Atendida'),

('No asistio'),

('Cancelada');

go

create table expediente

(

id\_expediente int IDENTITY(1,1) not null,

id\_tipo\_exp int not null,

fecha\_expediente datetime not null,

descripcion\_Expediente varchar(500) not null,

primary key (id\_expediente),

CONSTRAINT fk\_tipo\_exo

FOREIGN KEY ( id\_tipo\_exp )

REFERENCES tipo\_expediente ( id\_tipo\_exp)

);

go

create table cita

(

id\_cita int IDENTITY(1,1)not null,

id\_expediente\_cita int null,

fecha\_cita datetime not null,

id\_empleado\_cita int,

id\_paciente int ,

primary key (id\_cita),

CONSTRAINT fk\_cita\_estudiante

FOREIGN KEY ( id\_paciente)

REFERENCES estudiante ( id\_estudiante),

CONSTRAINT fk\_cita\_expediente

FOREIGN KEY ( id\_expediente\_cita)

REFERENCES expediente ( id\_expediente),

CONSTRAINT fk\_empleado\_cita

FOREIGN KEY ( id\_empleado\_cita)

REFERENCES empleado ( id\_empleado)

);

go

CREATE TABLE HistoricoCita(

id\_HistoricoCita int IDENTITY(1,1)not null,

id\_cita\_ int not null,

descripcion\_anotacion varchar (200) not null,

fecha\_modificacion datetime not null,

id\_empleado\_historico int,

id\_estadocita int,

primary key (id\_HistoricoCita),

CONSTRAINT fk\_cita\_historio

FOREIGN KEY ( id\_cita\_)

REFERENCES cita ( id\_cita),

CONSTRAINT fk\_estado\_cita

FOREIGN KEY ( id\_estadocita )

REFERENCES estado\_cita ( id\_estado\_cita),

CONSTRAINT fk\_cita\_historio\_empleado

FOREIGN KEY ( id\_empleado\_historico)

REFERENCES empleado ( id\_empleado)

);

go

## **Diccionario de Datos**

Tabla: Acudiente: Almacenara los datos del acudiente de cada estudiante

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_acudiente | int | 4 | NO | SI |  | 1,2,3 |
| dependencia\_econo\_ | varchar | 2 | NO |  |  | SI NO |
| nombre\_acudiente | varchar | 100 | NO |  |  | CARLOS |
| apellidos\_acudiente | varchar | 100 | NO |  |  | GONZALEZ |
| ocupacion\_acudiente | varchar | 100 | NO |  |  | COMERCIANTE |
| direccion\_acudiente | varchar | 200 | NO |  |  | CRA 58 05- 85 |
| departamento\_acudiente | int | 4 | NO |  | SI | CASANARE |
| telefono\_acudiente | varchar | 20 | NO |  |  | 320235841 |
| parentezo\_acudiente | varchar | 100 | NO |  |  | TIO |
| correo\_acudiente | varchar | 100 | NO |  |  | CARLOS GONZALEZ |

Tabla: Acudiente\_Estudiante: Almacenará los ids de cada estudiante por cada acudiente

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_acudiente\_estudainte | int | 4 | NO | SI |  | 1,2,3 |
| id\_estudiante | int | 4 | NO |  | SI | 1,2,3 |
| id\_acudiente | int | 4 | NO |  | SI | 1,2,3 |

Tabla: Archivo\_Estudiante: Almacenara los datos de los archivos adjuntos de cada estudiante

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_archivo | int | 4 | NO | SI |  | 1,2,3 |
| id\_estudiante\_archi | int | 4 | NO |  | SI | 1,2,3 |
| descripcion\_archivo | varchar | 100 | NO |  |  | DOCUMENTO  IDENTIDAD |
| direccion\_archivo | varchar | 500 | NO |  |  | URL |

Tabla: Cargo : Almacenara los cargos de la CRU

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_cargo | int | 4 | NO | SI |  | 1,2,3 |
| valor\_cargo | varchar | 20 | NO |  |  | Director |

Tabla Departamento: Almacenara los departamentos de Colombia

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_departamento | int | 4 | NO | SI |  | 1,2,3 |
| descripcion\_departamento | varchar | 100 | NO |  |  | BOLIVAR |

Tabla Cita: Almacenara los datos de las citas registradas

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_cita | int | 4 | NO | SI |  | 1,2,3, |
| id\_expediente\_cita | int | 4 | SI |  | SI | 1,2,3, |
| fecha\_cita | datetime | 8 | NO |  |  | 5/09/2000 |
| id\_empleado\_cita | int | 4 | SI |  | SI | 1,2,3, |
| id\_paciente | int | 4 | SI |  | SI | 1,2,3, |

Tabla Contrato: Almacenara los datos del contrato de cada empleado

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_contrato | int | 4 | NO | SI |  | 1,2,3, |
| id\_empleado | int | 4 | NO |  | SI | 1,2,3, |
| cargo\_contrato | int | 4 | NO |  | SI | 1,2,3, |
| fecha\_inicio | date | 3 | NO |  |  | 5/09/2000 |
| fecha\_fin | date | 3 | NO |  |  | 5/09/2000 |
| tipo\_de\_contrato | varchar | 50 | NO |  |  | INDEFINIDO |
| salario | bigint | 8 | SI |  |  | 1200000 |

Tabla Datos universidad: Almacenara los datos de la universidad donde este el estudiante

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_datosuniversidad | int | 4 | NO | SI |  | 1,2,3, |
| Universidad\_estudiante | varchar | 200 | NO |  |  | UDC |
| Facultad\_estudiante | varchar | 200 | NO |  |  | INGENIERIA |
| Programa\_estudiante | varchar | 200 | NO |  |  | SISTEMAS |
| Puntaje\_Basico\_Matricula | varchar | 20 | NO |  |  | 600.80 |
| Promedio\_Academico\_estudiante | varchar | 20 | SI |  |  | 4.2 |
| Año\_ingreso\_Universidad\_ | datetime | 8 | NO |  |  | 5/09/2000 |
| semestre\_ingreso\_universidad | int | 4 | NO |  |  | 5 |
| Porcentaje\_Avance | varchar | 2 | SI |  |  | 60 |
| id\_estudiante\_datosuniversidad | int | 4 | NO |  | SI | 1,2,3, |

Tabla Persona: Almacenara datos generales del estudiante y empleado

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | | Null | | PK | | FK | | Ejemplo | |
| id\_persona | int | 4 | | NO | | SI | |  | | 1,2,3 | |
| correo\_directorio | varchar | 300 | | NO | |  | |  | | director@cru.com | |
| contrasena | varchar | | 50 | | NO | |  | |  | | 123456 |
| id\_rol\_Directorio | int | 4 | | NO | |  | | SI | | 1,2,3 | |
| TipoIdentificacion | int | 4 | | NO | |  | | SI | | 1,2,3 | |
| NumeroIdentificacion | varchar | 15 | | NO | |  | |  | | 258963252 | |
| Nombres | varchar | 100 | | NO | |  | |  | | Carlos | |
| Apellidos | varchar | 100 | | NO | |  | |  | | González | |
| Estadocivil | int | 4 | | NO | |  | | SI | | 1,2,3 | |
| Estrato | int | 4 | | NO | |  | |  | | 1 | |
| Direccion | varchar | 100 | | NO | |  | |  | | Cll 5 No 8 | |
| Telefono | varchar | 20 | | NO | |  | |  | | 5289634 | |
| Tipo\_sangre | int | 4 | | NO | |  | | SI | | 1,2,3 | |
| Fechanacimiento | datetime | 8 | | NO | |  | |  | | 1/01/2000 | |
| MunicipioNacimiento | varchar | 100 | | NO | |  | |  | | Mesitas | |
| DepartamentoNacimiento | int | 4 | | NO | |  | | SI | | 1 | |
| Paisnacimiento | int | 4 | | NO | |  | | SI | | 2 | |

Tabla Elemento: Almacenara los datos de los elementos que tengan en cada habitación.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_elemento | int | 4 | NO | SI |  | 1,2,3 |
| id\_tipo\_elemento\_ | int | 4 | NO |  | SI | 1,2,3 |
| descripcion | varchar | 50 | NO |  |  | ESTUFA |
| marca | varchar | 50 | SI |  |  | HACEB |
| Modelo | varchar | 50 | SI |  |  | 2016 |

Tabla Empleado: Almacenara los ids de los empleados registrados en la CRU

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_empleado | int | 4 | NO | SI |  | 1,2,3 |
| empleado\_directorio | int | 4 | NO |  | SI | 1,2,3 |

Tabla EstadoCita: Almacenara los estados de la cita

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_estado\_cita | int | 4 | NO | SI |  | 1,2,3 |
| valor\_estado\_cita | varchar | 50 | NO |  |  | Agendada |

Tabla Espacio: Almacenara los datos de los espacios de la CRU

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_espacio | int | 4 | NO | SI |  | 1,2,3 |
| descripcion\_espacio | varchar | 100 | NO |  |  | 305 |
| capacidad | int | 4 | SI |  |  | 3 |
| cupo | int | 4 | SI |  |  | 3 |
| id\_tipo\_espacio\_ | int | 4 | NO |  | SI | 1,2,3 |
| id\_piso\_espacio | int | 4 | NO |  | SI | 1,2,3 |

Tabla Estado Espacio: Almacena los estados de un espacio habitacional

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_estado\_espacio | int | 4 | NO | SI |  | 1,2,3 |
| valor\_estado\_espacio | varchar | 10 | NO |  |  | Disponible |

Tabla Estado Solicitud: Almacena los estados de una solicitud

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_estado\_solciitud | int | 4 | NO | SI |  | 1,2,3 |
| valor\_estado\_solciitud | varchar | 50 | NO |  |  | Registrada |

Tabla Estado Usuario: Almacena los estados de un usuario en la CRU

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| Id\_estado\_usuario\_cru | Int | 4 | NO | SI |  | 1,2,3 |
| Descripcion\_estado\_usuario\_cru | Varchar | 30 | NO |  |  | Activo |

Tabla Estado Civil: Almacena los estados civiles de los empleados y estudiantes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_estadocivil | int | 4 | NO | SI |  | 1,2,3 |
| valor\_estadocivil | varchar | 20 | NO |  |  | soltero |

Tabla Estudiante: Almacena datos de los estudiantes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_estudiante | int | 4 | NO | SI |  | 1,2,3 |
| Servicio\_Salud | varchar | 200 | NO |  |  | Famisanar |
| Dispacidad\_estudiante | varchar | 2 | NO |  |  | NO |
| descripcion\_dispacacidad\_estudainte | varchar | 200 | SI |  |  | NINGUNA |
| Situaciondesplazamientoestudiante | varchar | 2 | NO |  |  | NO |
| Numerohermanos | int | 4 | NO |  |  | 2 |
| tipodevivienda\_estudiante | int | 4 | NO |  | SI | 1,2,3 |
| apoyouniversidad | varchar | 2 | NO |  |  | SI |
| descripcion\_apoyo\_ | varchar | 100 | SI |  |  | Almuerzo |
| raza\_estudiante | int | 4 | NO |  | SI | 1,2,3 |
| id\_espacio\_estudiante | int | 4 | SI |  | SI | 1,2,3 |
| id\_directorio\_estudiante | int | 4 | NO |  | SI | 1,2,3 |

Tabla Expediente: Almacena el registro de la cita psicológica de cada cita

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_expediente | int | 4 | NO | SI |  | 1,2,3 |
| id\_tipo\_exp | int | 4 | NO |  | SI | 1,2,3 |
| fecha\_expediente | datetime | 8 | NO |  |  | 2/05/2017 |
| descripcion\_Expediente | varchar | 500 | NO |  |  | El paciente reporta… |

Tabla Histórico Directorio: Almacena las modificaciones que tenga el estudiante o el empleado

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_historico\_directorio | int | 4 | NO | SI |  | 1,2,3 |
| id\_directorio\_ | int | 4 | NO |  | SI | 1,2,3 |
| Estado\_directorio\_CRU | int | 4 | NO |  | SI | 1,2,3 |
| descripcion\_histoico | varchar | 200 | NO |  |  | Cambio de dirección |
| fecha | date | 3 | NO |  |  | 5/08/2017 |
| id\_empleado | int | 4 | SI |  | SI | 1,2,3 |

Tabla Histórico Espacio: Almacena las modificaciones que tenga el espacio

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_historico\_espacio | int | 4 | NO | SI |  | 1,2,3 |
| fecha\_historico\_espacio | date | 3 | NO |  |  | 5/08/2017 |
| estado\_espacio\_o | int | 4 | NO |  | SI | 1,2,3 |
| id\_espacio\_historico | int | 4 | NO |  | SI | 1,2,3 |
| id\_empleado\_historico | int | 4 | SI |  | SI | 1,2,3 |

Tabla Histórico Solicitud: Almacena las modificaciones que tenga la solicitud

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_historico\_solicitud | int | 4 | NO | SI |  | 1,2,3 |
| id\_caso\_anotacion | int | 4 | NO |  | SI | 1,2,3 |
| descripcion\_anotacion | varchar | 200 | NO |  |  |  |
| fecha\_modificacion | datetime | 8 | NO |  |  |  |
| id\_empleado\_historico | int | 4 | SI |  | SI | 1,2,3 |
| id\_estado\_solicitud\_ | int | 4 | SI |  | SI | 1,2,3 |

Tabla Historico Cita: Almacena las modificaciones que tenga la cita

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_HistoricoCita | int | 4 | NO | SI |  | 1,2,3 |
| id\_cita\_ | int | 4 | NO |  | SI | 1,2,3 |
| descripcion\_anotacion | varchar | 200 | NO |  |  | Tengo … |
| fecha\_modificacion | datetime | 8 | NO |  |  | 5/08/2017 |
| id\_empleado\_historico | int | 4 | SI |  | SI | 1,2,3 |
| id\_estadocita | int | 4 | SI |  | SI | 1,2,3 |

Tabla Historico Estudiante: Almacena las modificaciones que tenga el estudiante

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_historico\_ | int | 4 | NO | SI |  | 1,2,3 |
| id\_estudiante | int | 4 | NO |  | SI | 1,2,3 |
| fecha\_historico\_ | datetime | 8 | NO |  |  | 4/05/2017 |
| descripcion\_historico\_ | varchar | 500 | NO |  |  | Cambio Apoyo  Universidad |
| id\_empleado\_ | int | 4 | SI |  | SI | 1,2,3 |

Tabla Histórico Inventario: Almacena las modificaciones de cantidad que se hagan en el inventario

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_inventario\_ | int | 4 | NO | SI |  | 1,2,3 |
| id\_empleado\_historico\_ | int | 4 | NO |  | SI | 1,2,3 |
| descripcion\_historico\_ | varchar | 200 | NO |  |  | Se han agregado 3 unidades |
| fecha\_historico\_ | datetime | 8 | NO |  |  | 4/05/2014 |
| inventario\_id\_ | int | 4 | NO |  | SI | 1,2,3 |

Tabla Inventario Espacio: Almacenara los datos de los elementos que contenga un espacio

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_inventario\_esp | Int | 4 | NO | SI |  | 1,2,3 |
| id\_elemento\_inventario\_ | Int | 4 | NO |  | SI | 1,2,3 |
| cantidad\_inventario\_elemento\_ | Int | 4 | NO |  |  | 5 |
| fecha\_inventario\_ | Datetime | 8 | NO |  |  | 4/05/2017 |
| id\_espacio\_inventario\_\_ | Int | 4 | NO |  | SI | 1,2,3 |

Tabla Pais: Almacenara los datos de los paises

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | | Null | | PK | | FK | | Ejemplo | |
| id\_pais | int | 4 | | NO | | SI | |  | | 1,2,3 | |
| descripcionPais | varchar | | 100 | | NO | |  | |  | | COLOMBIA | |

Tabla Permiso: Almacenara los permisos de cada modulo

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| Permiso\_id | int | 4 | NO | SI |  | 1,2,3 |
| Modulo | varchar | 50 | NO |  |  | EMPLEADOS |
| descripcion | varchar | 50 | NO |  |  | PUEDE MODIFICAR |

Tabla Permiso\_Rol: Almacena los permisos de cada rol del sistema

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| Rol\_id | int | 4 | NO |  | SI | 1,2,3 |
| Permiso\_id\_ | int | 4 | NO |  | SI | 1,2,3 |

Tabla Piso: Almacena los pisos del edificio de la cru

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_piso | int | 4 | NO | SI |  | 1,2,3 |
| valor\_piso | varchar | 2 | NO |  |  | 1 |

Tabla Prioridad Solicitud: Almacena las prioridades de las solicitudes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_prioridad\_solciitud | Int | 4 | NO | SI |  | 1,2,3 |
| valor\_prioridad\_solciitud | Varchar | 50 | NO |  |  | ALTA |
| tiempodesolucion\_horas | Int | 4 | NO |  |  | 6 HORAS |

Tabla Raza: Almacena los datos de las razas

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_raza | Int | 4 | NO | SI |  | 1,2,3 |
| descripcion\_raza | varchar | 14 | NO |  |  | MULATO |

Tabla Reparación Espacio: almacena los datos de las reparaciones solicitadas

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_reparacion | int | 4 | NO | SI |  | 1,2,3 |
| fecha\_reparacion\_espacio | datetime | 8 | NO |  |  | 5/06/2017 |
| descripcion\_reparacion | varchar | 500 | NO |  |  | Se realiza cambio de Toma |
| espacio\_reparacion | int | 4 | NO |  | SI | 1,2,3 |
| costo | bigint | 8 | NO |  |  | 20000 |
| id\_solicitud\_reparacion\_ | int | 4 | NO |  | SI | 1,2,3 |

Tabla Rol: Almacena los datos de los roles

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_rol | int | 4 | NO | SI |  | 1,2,3 |
| descripcion\_rol | varchar | 50 | NO |  |  | Administrador |

Tabla Solicitud: Almacena los datos de las solicitudes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_solicitud | int | 4 | NO | SI |  | 1,2,3 |
| id\_tipo\_solicitud\_\_ | int | 4 | NO |  | SI | 1,2,3 |
| id\_prioridad\_solciitud\_\_ | int | 4 | NO |  | SI | 1,2,3 |
| descripcion\_solicitud | varchar | 200 | NO |  |  | Se dañó la toma |
| id\_estudiante\_solicitud | int | 4 | NO |  | SI | 1,2,3 |
| id\_empleado\_solicitud | int | 4 | NO |  | SI | 1,2,3 |

Tabla Solución: Almacena los datos de las soluciones de las solicitudes y reparaciones

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_solucion | int | 4 | NO | SI |  | 1,2,3 |
| descripcion\_solucion | varchar | 100 | NO |  |  | Se realizó cambio de toma |
| id\_empleado\_solucion | int | 4 | SI |  | SI | 1,2,3 |
| fecha\_solucion | datetime | 8 | SI |  |  | 5/05/2017 |
| id\_solicitud\_solucion | int | 4 | NO |  | SI | 1,2,3 |

Tabla Tipo elemento: Almacena los datos de los tipos de elementos de la CRU

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_tipo\_elemento | int | 4 | NO | SI |  | 1,2,3 |
| valor\_tipo\_elemento | varchar | 30 | NO |  |  | Eléctrico |

Tabla Tipo Espacio: Almacena el tipo de espacio

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_tipo\_espacio | int | 4 | NO | SI |  | 1,2,3 |
| valor\_tipo\_espacio | varchar | 20 | NO |  |  | Residencial |

Tabla Tipo Expediente

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_tipo\_exp | int | 4 | NO | SI |  | 1,2,3 |
| valortipoexp | varchar | 50 | NO |  |  | Psicológico |

Tabla Tipo Solicitud: Almacena el tipo de solicitud a registrar

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_tipo\_solicitud | int | 4 | NO | SI |  | 1,2,3 |
| valor\_tipo\_solicitud | varchar | 50 | NO |  |  | Administrativa |

Tabla Tipo de Sangre: Almacena los tipos de sangre

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_tipo\_sangre | int | 4 | NO | SI |  | 1,2,3 |
| descripcion\_tipo\_sangre | varchar | 11 | NO |  |  | O+ |

Tabla Tipo de Vivienda: Almacena los tipos de vivienda

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_tipodevivienda | int | 4 | NO | SI |  | 1,2,3 |
| tipovivienda | varchar | 9 | NO |  |  | Propia |

Tabla Tipo de Identificación: Almacena los tipos de identificación

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Tamaño | Null | PK | FK | Ejemplo |
| id\_tipoidentificacion | int | 4 | NO | SI |  | 1,2,3 |
| tipoidentificacion | varchar | 50 | NO |  |  | Tarjeta de Identidad |

1. **DEFINICION DE ACTORES Y ROLES**

|  |  |
| --- | --- |
| **Actor** | **ACT. 001 Director** |
| **Descripción** | Corresponde al máximo cargo de la CRU |
| **Responsabilidades** | * Ingreso seguro al sistema * Registro, consulta y actualización de información de **Empleados** * Consulta y actualización de estado de **admisiones** * Consulta y actualización de información de **expedientes** **de estudiantes** * Registro, Consulta y modificación de información de **espacios** físicos. * Consulta de información de **reparaciones de los espacios** físicos y **citas** * Registro, Consulta y modificación de información de **inventario** de elementos de cada espacio * Consulta y modificación de información de **solicitudes administrativas** |
| **Fuentes** | José Primo, Director Ejecutivo |

|  |  |
| --- | --- |
| **Actor** | **ACT. 002 Psicólogo** |
| **Descripción** | Corresponde a empleado de la CRU |
| **Responsabilidades** | * Ingreso seguro al sistema * Consulta y actualización de información de expedientes de estudiantes. * Consulta de espacios y sus respectivos inventarios asignados a estudiantes. * Consulta y modificación de información de **solicitudes** **administrativas** y **citas psicológicas** |
| **Fuentes** | José Primo, Director Ejecutivo |

|  |  |
| --- | --- |
| **Actor** | **ACT. 003 Estudiante** |
| **Descripción** | Corresponde a los residentes de la CRU |
| **Responsabilidades** | * Ingreso seguro al sistema * Registro y consulta de información de admisión * Consulta y actualización de información de expediente propio * Registra y Consulta de información de solicitudes administrativas, reparaciones y citas psicológicas. |
| **Fuentes** | José Primo, Director Ejecutivo |

|  |  |
| --- | --- |
| **Actor** | **ACT. 004 Mantenimiento** |
| **Descripción** | Corresponde a empleado que realiza las reparaciones en la CRU |
| **Responsabilidades** | * Ingreso seguro al sistema * Consulta de espacios * Consulta de información de contacto del estudiante * Consulta y modificaciones reparaciones de espacios. * Consulta de inventarios de Espacios |
| **Fuentes** | José Primo, Director Ejecutivo |

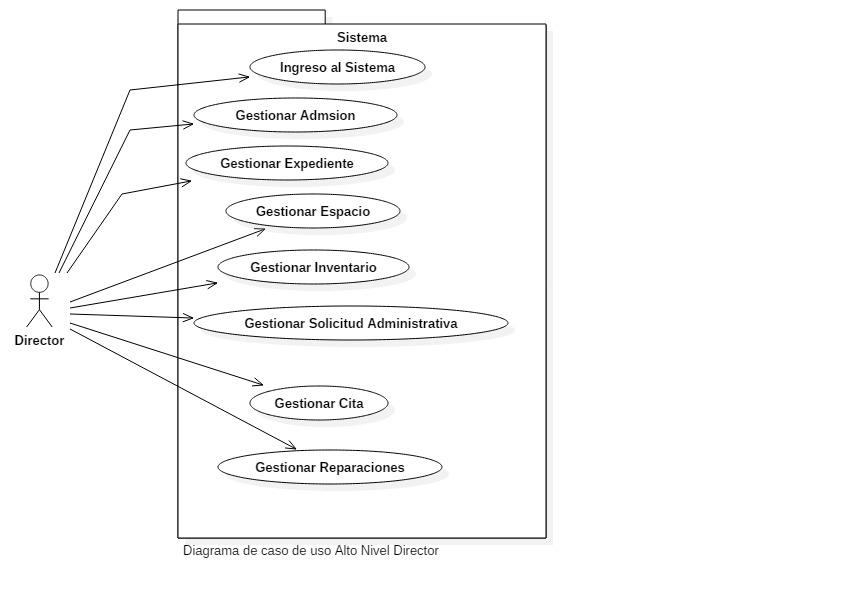
1. **CASOS DE USO**
   1. **Caso de Uso Alto Nivel**
      1. **Director**

Ilustración 3 Diagrama de caso de uso de alto nivel Director

* + 1. **Estudiante**

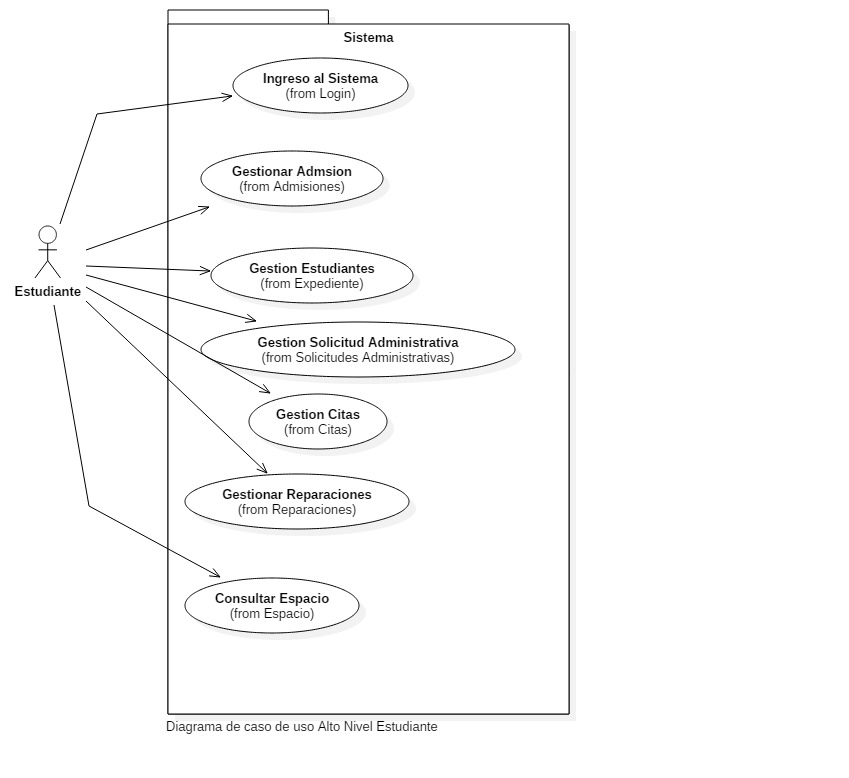
****

Ilustración 4 Diagrama de caso de uso de alto nivel Estudiante

* + 1. **Psicólogo**

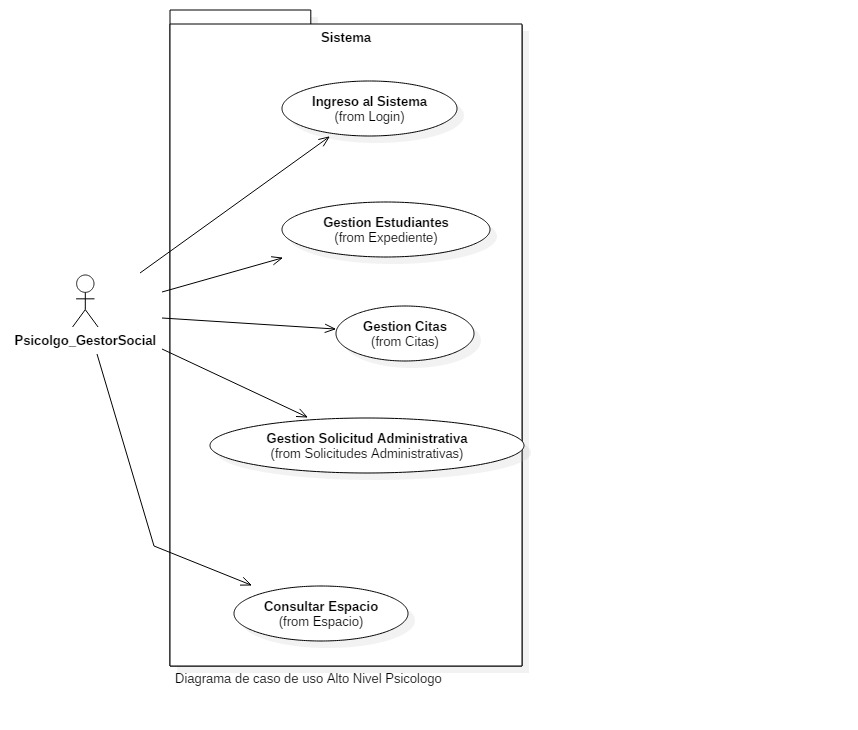
****

Ilustración 5 Diagrama de caso de uso de alto nivel Psicólogo

* + 1. **Mantenimiento**

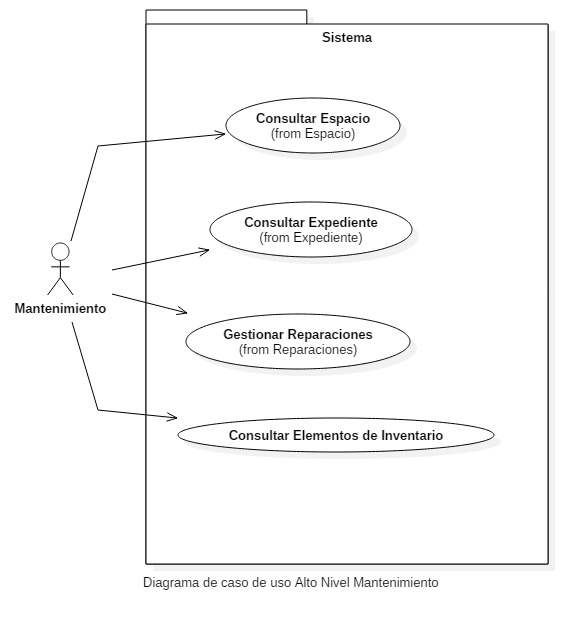
****

Ilustración 6 Diagrama de caso de uso de alto nivel Mantenimiento

* 1. **Caso de Uso Extendido**
     1. **Gestión de Empleado**

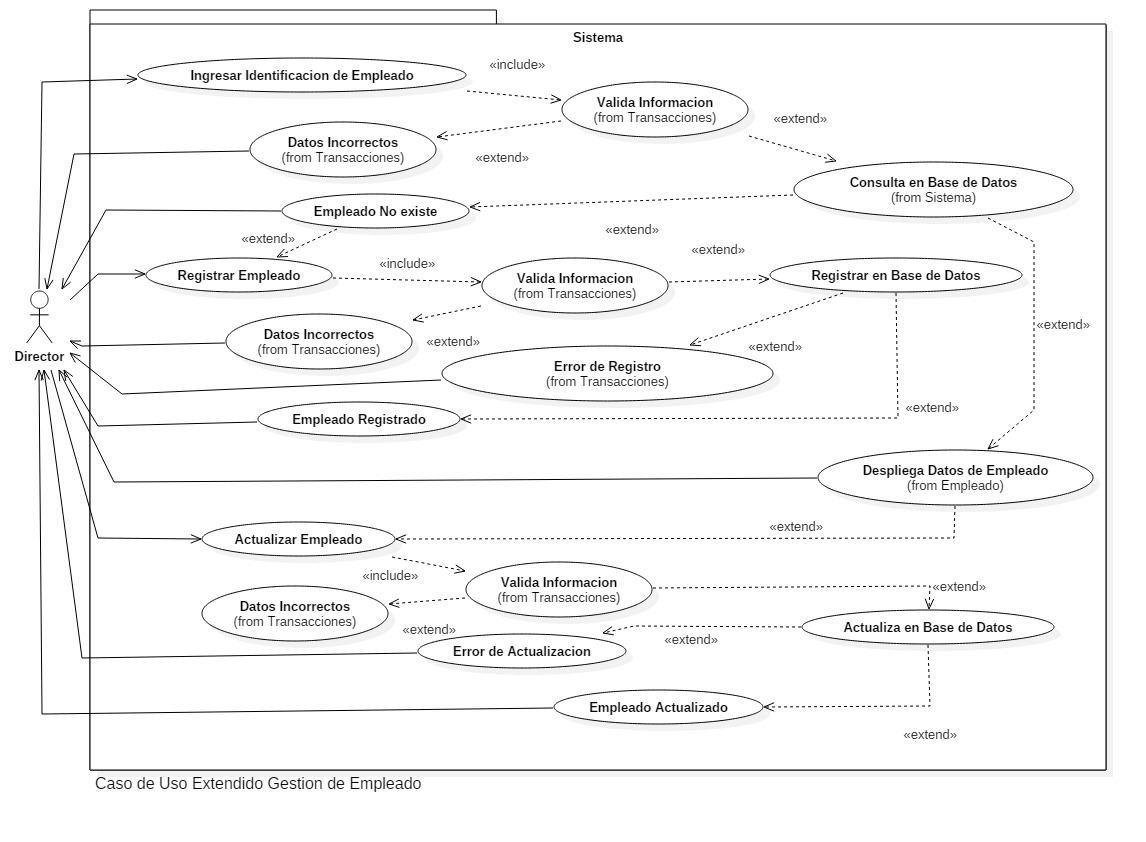
****

Ilustración 7 Diagrama de caso de uso de extendido Gestión de Empleado

* + 1. **Gestión de Admisiones**

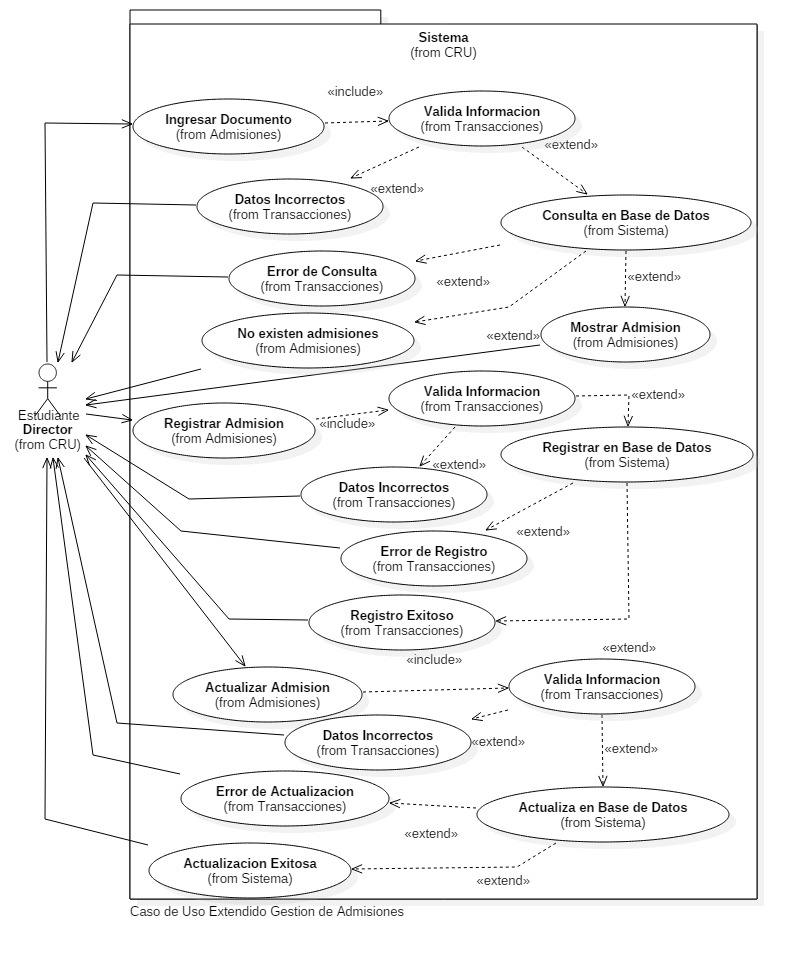
****

Ilustración 8 Diagrama de caso de uso de extendido Gestión de Admisiones

* + 1. **Gestión de Citas – Estudiante**

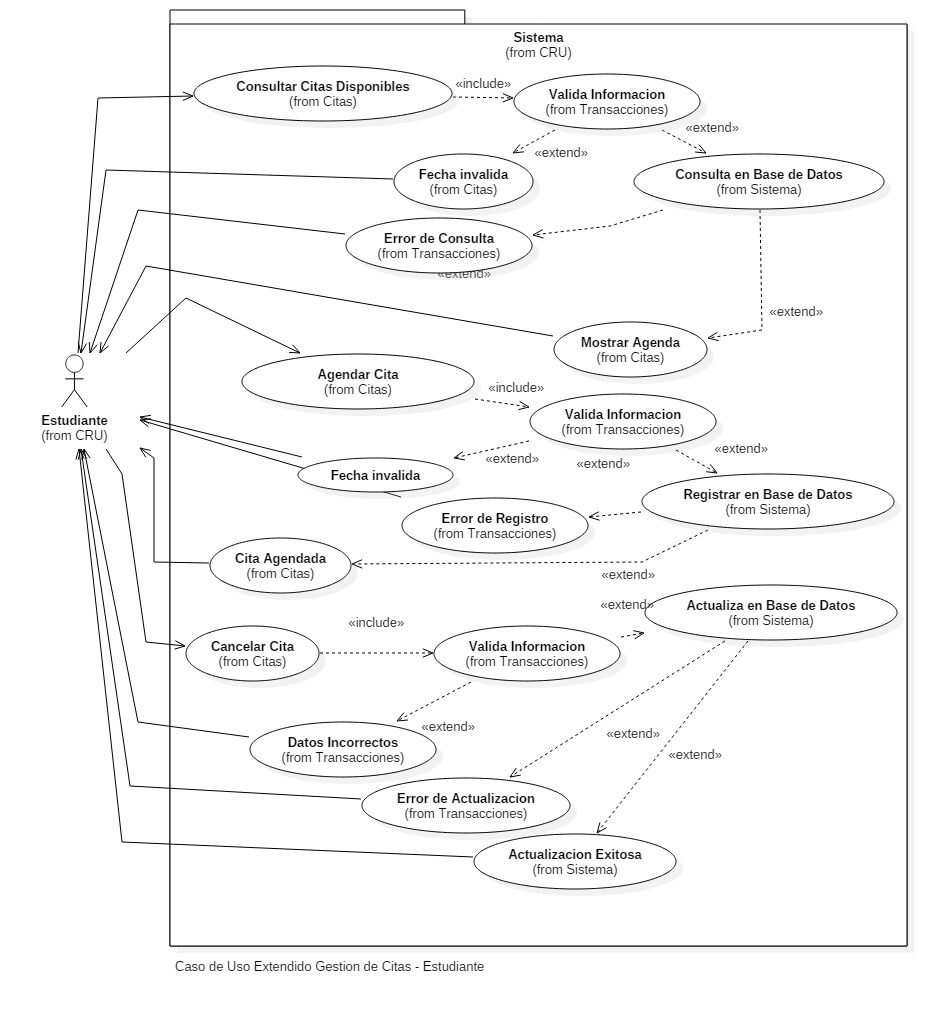
****

Ilustración 9 Diagrama de caso de uso de extendido Gestión de Citas Estudiante

* + 1. **Gestión de Citas – Psicólogo**

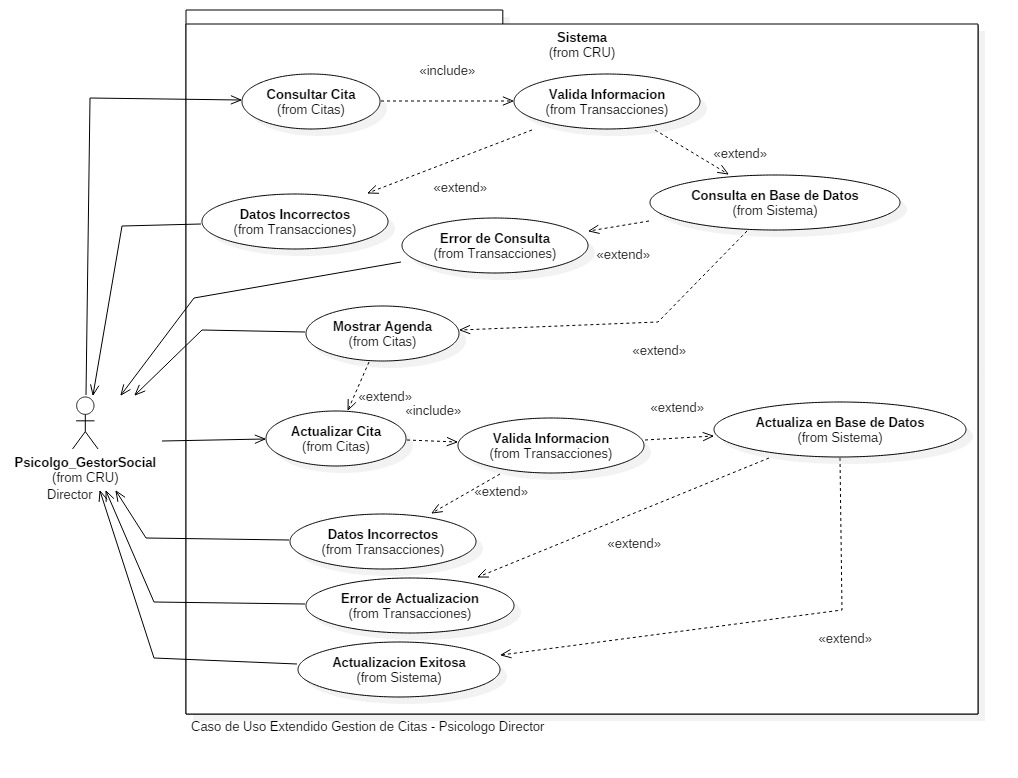
****

Ilustración 10 Diagrama de caso de uso de extendido Gestión de Citas Psicólogo

* + 1. **Gestión de Elementos**

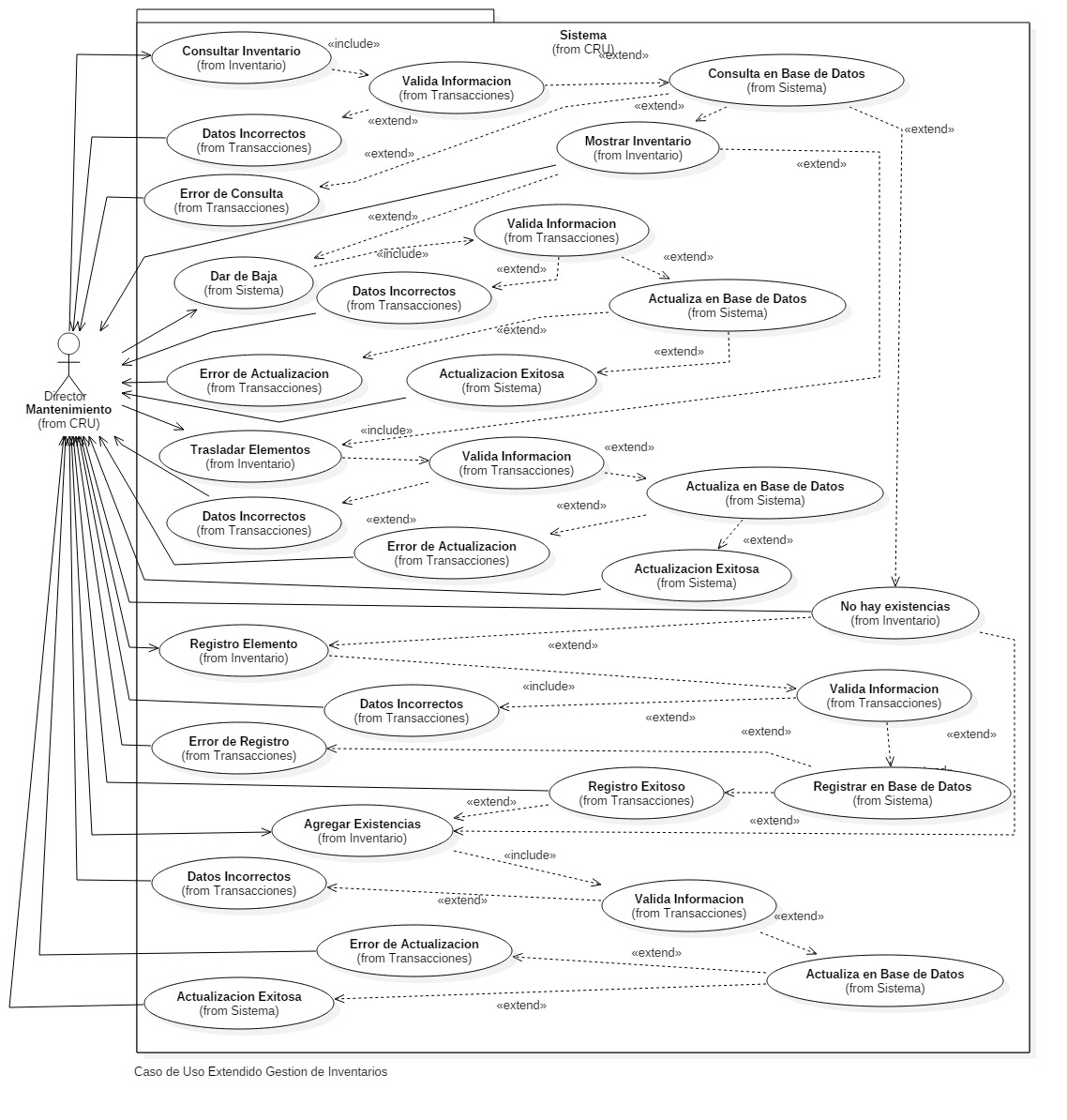
****

Ilustración 11 Diagrama de caso de uso de extendido Gestión de elementos

* + 1. **Gestión de Espacios**

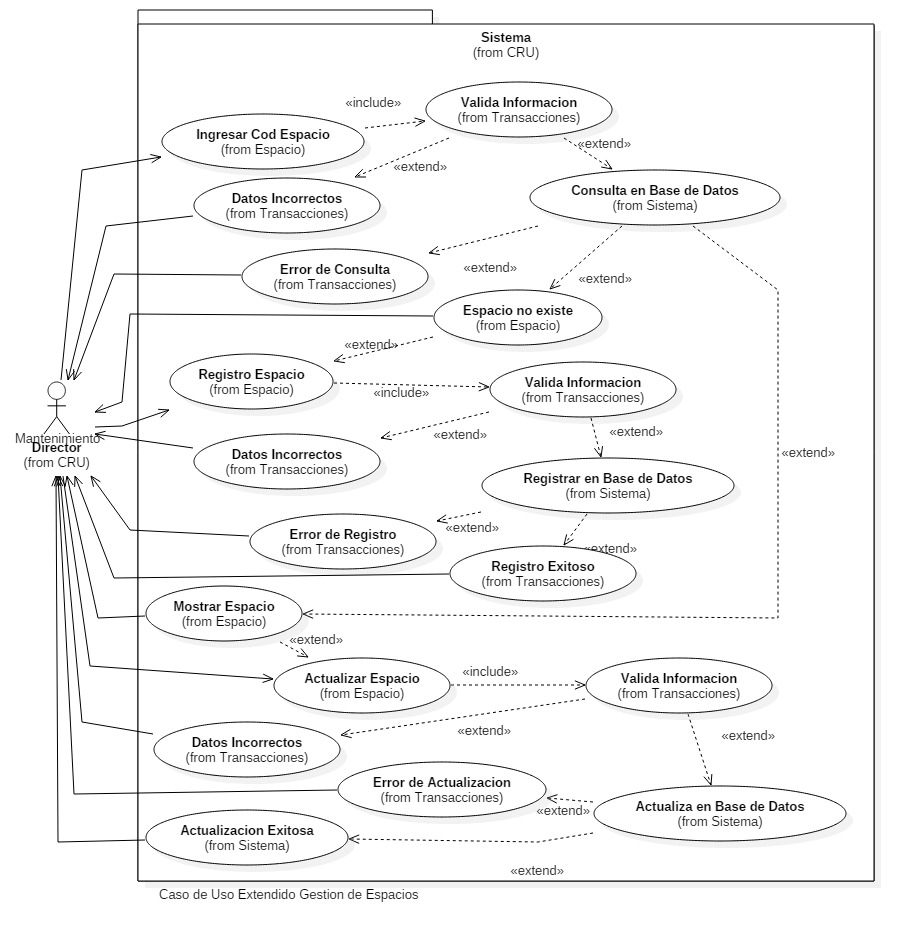
****

Ilustración 12 Diagrama de caso de uso de extendido Gestión de Espacios

* + 1. **Gestión de Estudiantes**

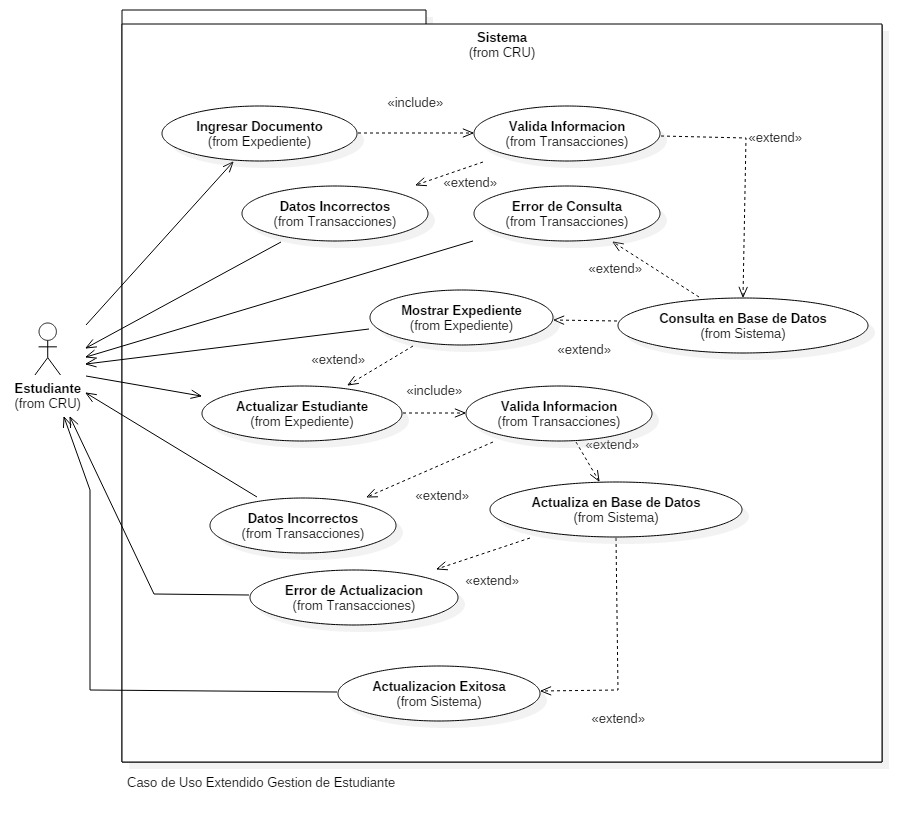


Ilustración 13 Diagrama de caso de uso de extendido Gestión de Estudiantes

* + 1. **Gestión de Reparaciones**

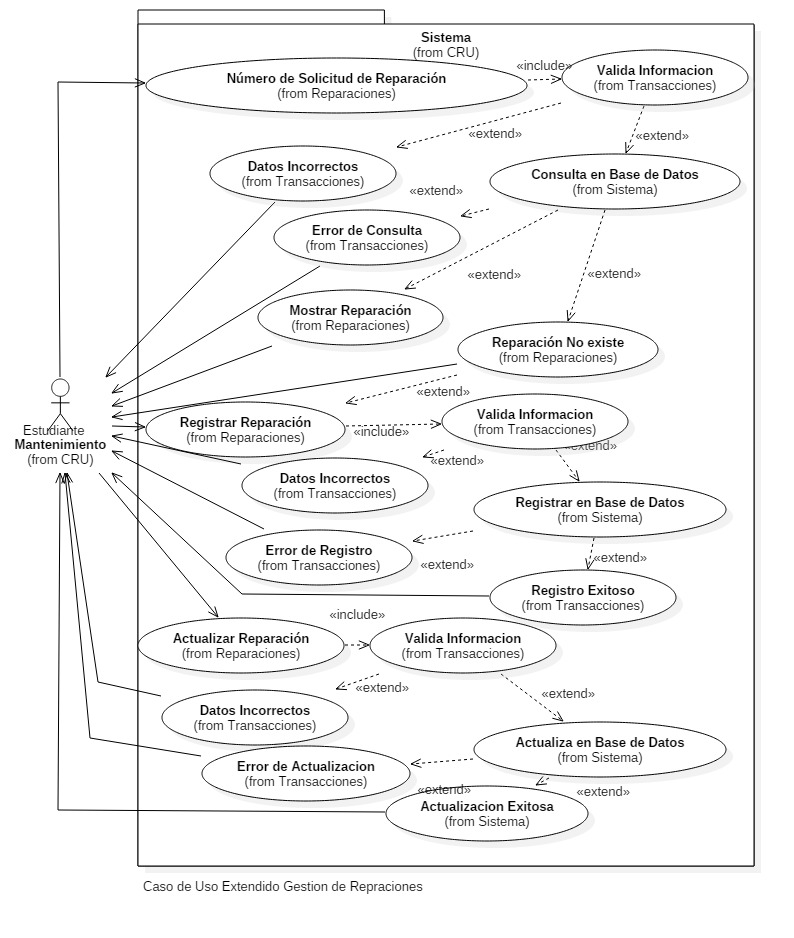
****

Ilustración 14 Diagrama de caso de uso de extendido Gestión de Reparaciones

* + 1. **Gestión de Solicitudes**

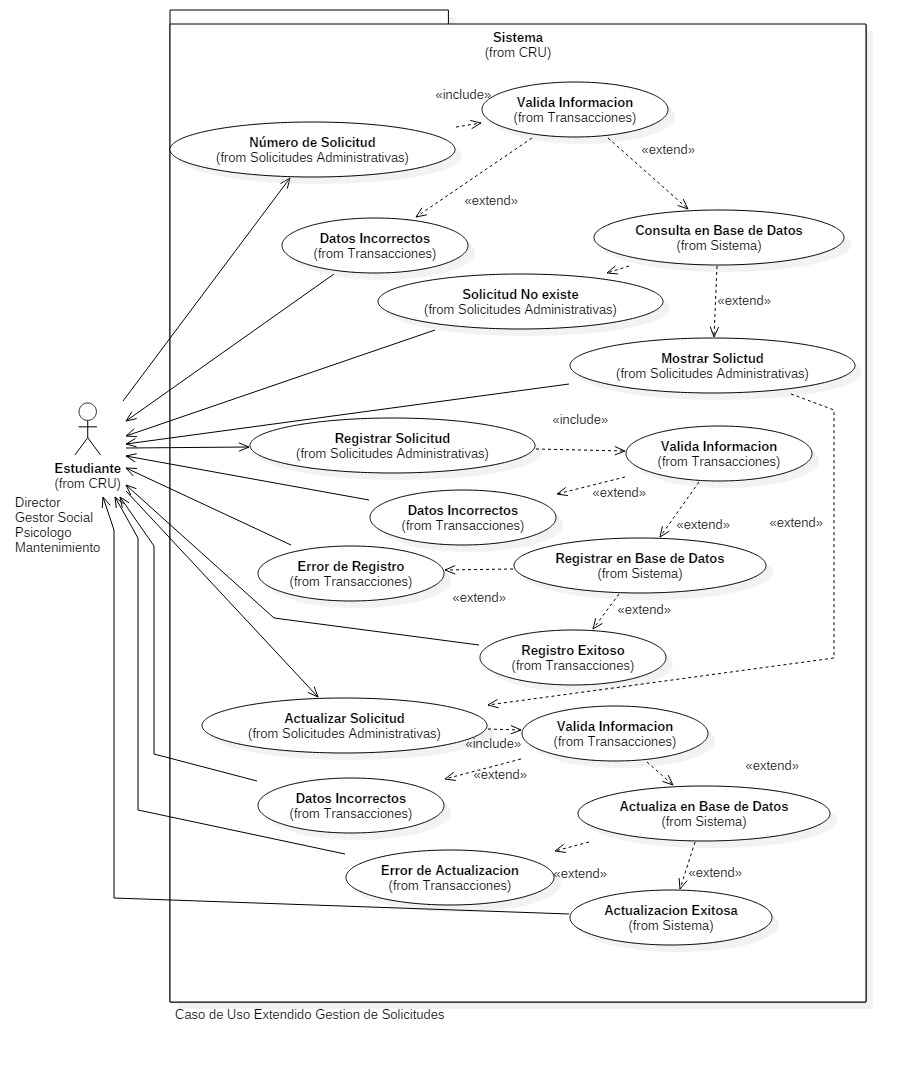
****

Ilustración 15 Diagrama de caso de uso de extendido Gestión de Solicitudes

1. **DIAGRAMA DE CLASES**
   1. **Diagrama de Clases**

**Ver Diagrama**

* 1. **Diagrama de Clases de Control**

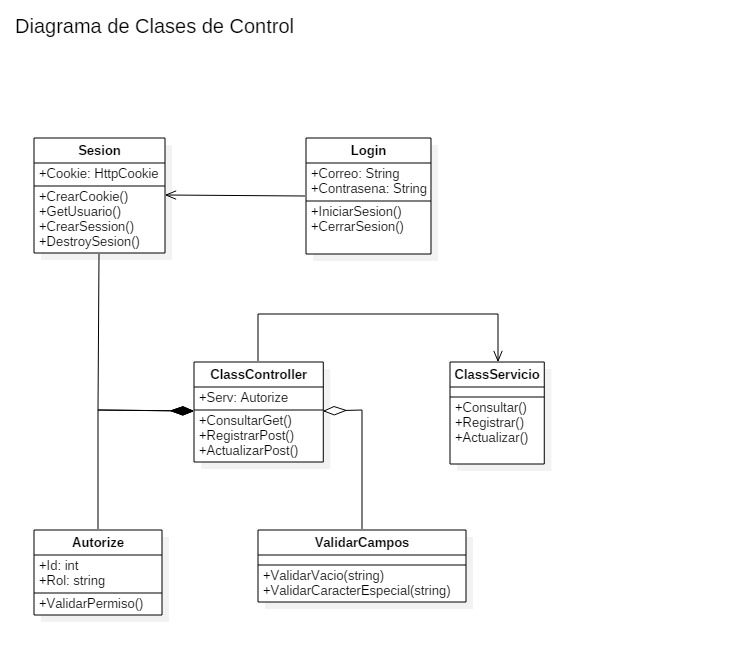
****

Ilustración 16 Diagrama de Clases de Control

* 1. **Diagrama de Clases de Navegación**

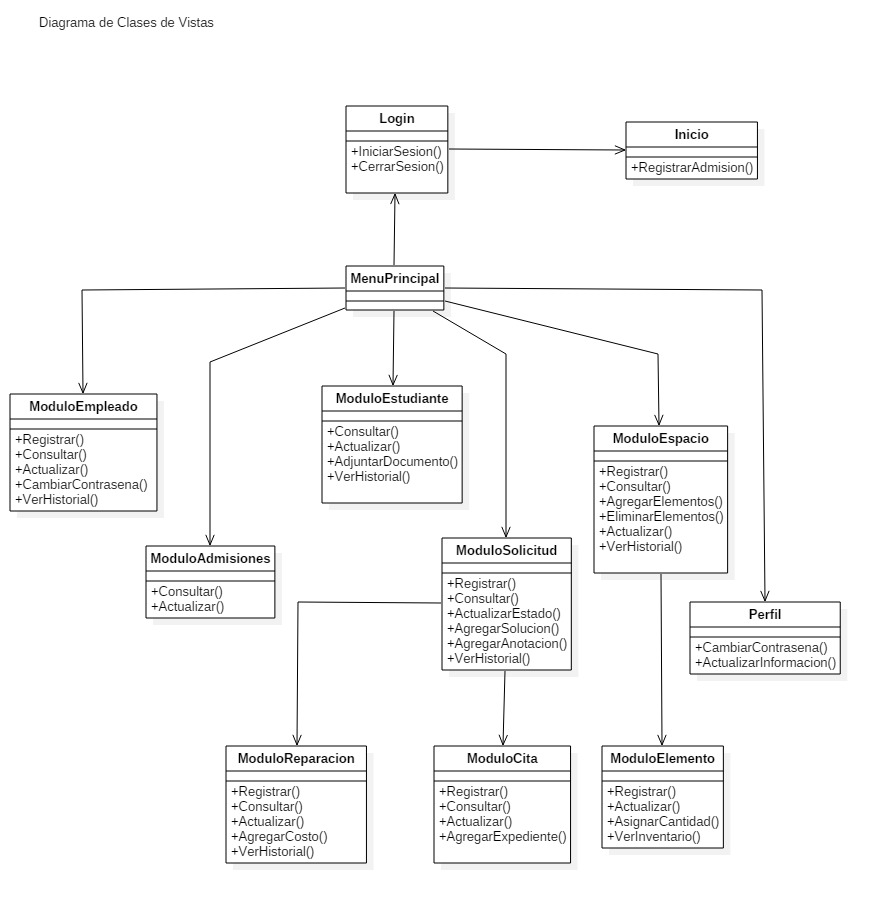
****

Ilustración 17 Diagrama de Clases de Navegación

1. **DIAGRAMA COMPONENTES**

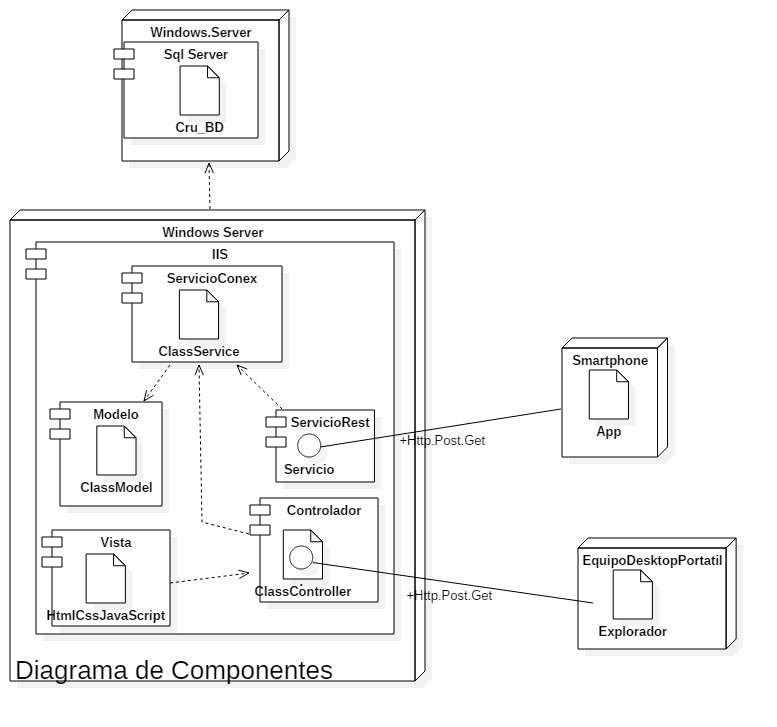
****

Ilustración 18 Diagrama de Componentes

1. **DIAGRAMA DE PAQUETES**

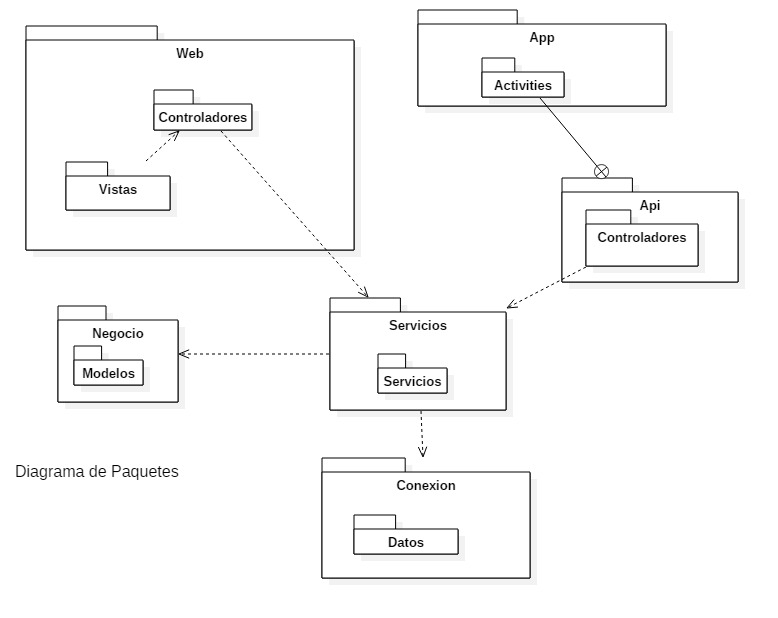
****

Ilustración 19 Diagrama de Paquetes

1. Ver anexo de encuesta estudiantes [↑](#footnote-ref-1)
2. Microsoft, “Plataforma de servicios de infraestructura”. Recuperado de: https://azure.microsoft.com/es-es/ [↑](#footnote-ref-2)