

Diseño de Arquitectura de Software

AdmisiónTEC

Versión 1.0

12 de Octubre 2023

Sponsor

Instituto Tecnológico de Costa Rica

Supervisora

Ericka Solano Fernández

Miembros del equipo

Hansol Antay

Elon Musk

Contacto

rostrhan@outlook.com

elon@musk.com

Índice

1	Introducción	3
1.1	Propósito	3
1.2	Alcance	3
1.3	Definiciones, acrónimos y abreviaciones	3
1.4	Referencias	3
2	Representación arquitectónica	3
3	Objetivos y restricciones de la arquitectura	5
3.1	Objetivos	5
3.2	Restricciones	5
4	Vista casos de uso	5
4.1	Caso de uso: Inicio de sesión	6
4.2	Caso de uso: Realizar solicitud de examen	7
4.3	Caso de uso: Simular notas de exámenes	9
4.4	Caso de uso: Asignar estados de solicitudes	10
4.5	Caso de uso: Generar reportes	11
4.6	Caso de uso: Gestionar cortes de carreras	13
5	Vista lógica	14
5.1	Modelo de negocios	14
5.2	Controladores y Vistas	17
5.3	Propuestas de vistas	17
6	Vista de deployment	19
7	Calidad	20
8	Tamaño y rendimiento	21

Historial de versiones

Fecha	#	Descripción	Autor
12/08/2023	1.0	Creación del documento base, es una entrega preliminar.	H. Antay

1. Introducción

1.1. Propósito

Este proyecto tiene como propósito principal brindar al ITCR una herramienta en línea que lo ayude a gestionar los procesos administrativos que conllevan el examen de admisión de la institución. Con este documento en particular se busca brindar una visión inicial del sistema modelando y detallando sus componentes, interacciones y funcionalidades principales.

1.2. Alcance

En este documento se va a cubrir las especificaciones arquitectónicas del sistema de solicitudes para el examen de admisión del ITCR. En este se va a cubrir la interacción con la bases de datos, modelado de clases, especificación del proceso de registro de solicitantes y el proceso de simulación y notificación de notas.

1.3. Definiciones, acrónimos y abreviaciones

- **TEC:** Tecnológico de Costa Rica.
- **ITCR:** Instituto Tecnológico de Costa Rica.
- **UML:** Unified Modeling Language, significa lenguaje de modelado unificado. [1]
- **MVC:** Patrón de diseño de software Modelo-Vista-Controlador.

1.4. Referencias

- [1] OMG. «Unified Modeling Language (OMG UML) Version 2.5.1.» (dic. de 2017), dirección: <https://www.omg.org/spec/UML/>.

2. Representación arquitectónica

En la especificación del documento se indica que al momento en el que se registra un usuario este va a tener que indicar la carrera en la que desea ingresar. Se consideró algo poco práctico por lo que esta propuesta tendrá un enfoque diferente.

Considerando que una persona puede realizar varios exámenes de admisión se planteó que el registro se hiciera sin la necesidad de indicar la carrera directamente. Se hace un registro "común", y ya una vez ingresado en el sistema el usuario podrá optar por realizar un examen de admisión según sea habilitado.

Esta propuesta sería más práctica. Pensando más a futuro en la evolución de la aplicación, el usuario podría seleccionar distintas fechas/horas para realizar la aplicación del examen, podría consultar de forma más rápida los resultados o estados de los exámenes de admisión, gestionar de forma más directa los pagos para poder realizar el examen e incluso se podría escalar y considerar exámenes de otras universidades. El diagrama de componentes de detalla a continuación:

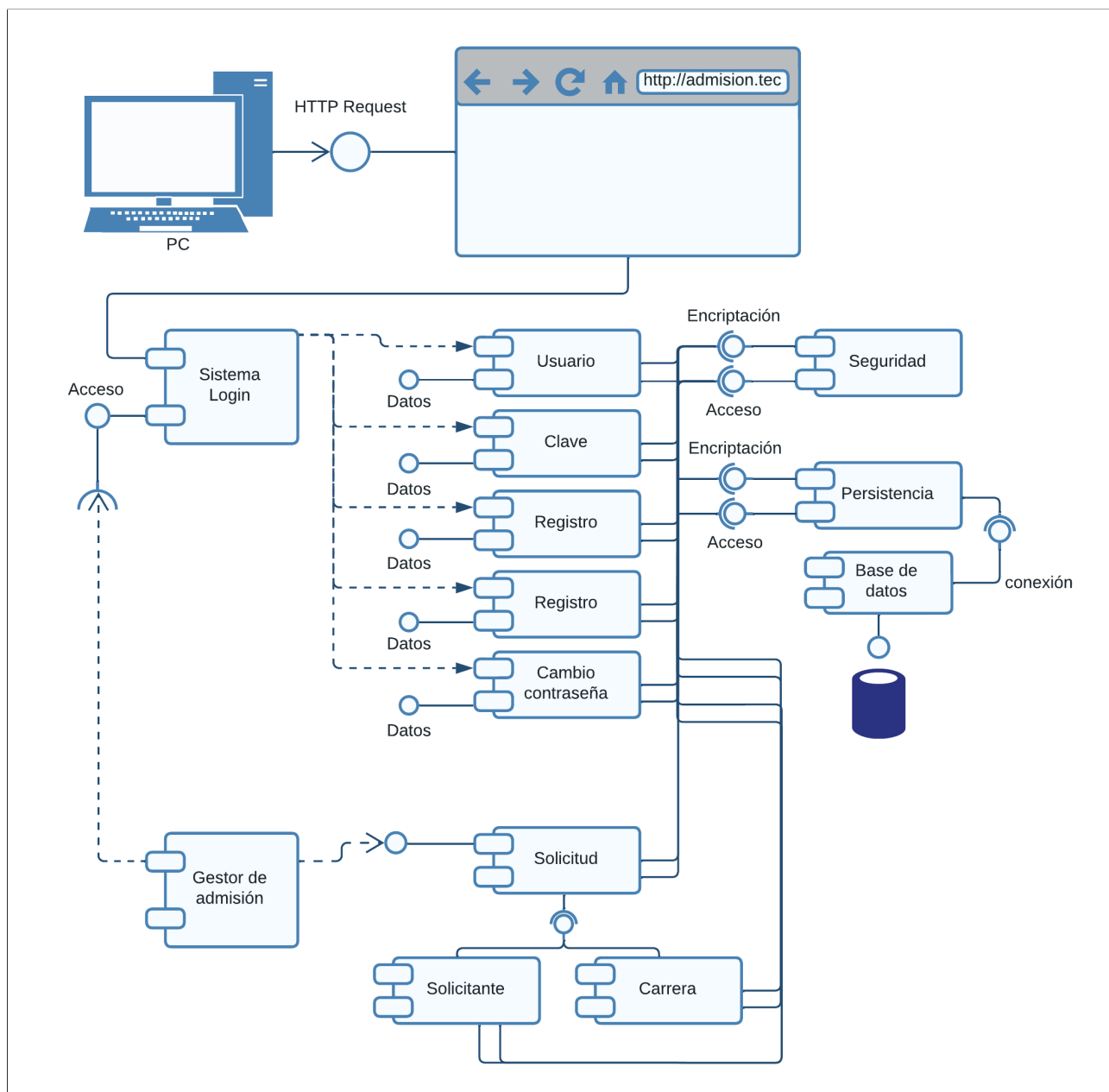


Figura 1: Diagrama de componentes

3. Objetivos y restricciones de la arquitectura

3.1. Objetivos

- **Escalabilidad.** Como se mencionó anteriormente, se podría integrar fácilmente más carreras e incluso instituciones que quieran hacer uso de este sistema.
- **Usabilidad.** Debe ser un sistema fácil de utilizar, con una interfaz sencilla pero agradable para los usuarios finales.
- **Seguridad.** La información debe manejarse de forma segura, todos los datos estarán cifrados y descifrados de forma segura.
- **Persistencia de datos.** Todos los datos deben almacenarse de forma persistente en una base de datos.
- **Disponibilidad.** El sistema deberá estar disponible en momentos de alta demanda. Considerando en períodos de tiempo en dónde se anuncia o promociona las solicitudes para la realización del examen de admisión y cuando se brindarán los resultados de los mismos.

3.2. Restricciones

- Todos los usuarios deberán identificarse para poder acceder a la plataforma. Este mecanismo debe ser completamente seguro.
- Los administradores podrán modificar los datos de *nota mínima de admisión* y la *cantidad máxima de gente admisible* en las carreras pero respetando restricciones.
- Los datos de solicitudes deberán enviarse de forma encriptada a la base de datos utilizando mecanismos de seguridad.
- El sistema deberá implementarse haciendo uso del patrón de diseño Modelo-Vista-Controlador.

4. Vista casos de uso

En esta sección veremos los casos de uso más relevantes para el sistema, primero definiremos textualmente estos casos de uso y los mostraremos con su respectivo diagrama.

4.1. Caso de uso: Inicio de sesión

Caso de uso: Inicio de sesión

Actor: Usuario

Precondiciones

1. El usuario está registrado.
2. El usuario ingresó a la página de inicio de sesión.

Flujo principal

1. El usuario ingresa usuario y contraseña.
2. El sistema valida la información.
3. Si la información es válida, se permite el acceso y se redirecciona a la página principal.

Flujo alternativo

2a. Información del usuario incorrecta.

- Se muestra un mensaje de error.
- El usuario puede volver a intentar.

2b. El usuario indica que olvidó la contraseña.

- El usuario hace clic en la opción "olvidé mi contraseña".
- Se realiza el caso de uso **Recuperación de cuenta**.

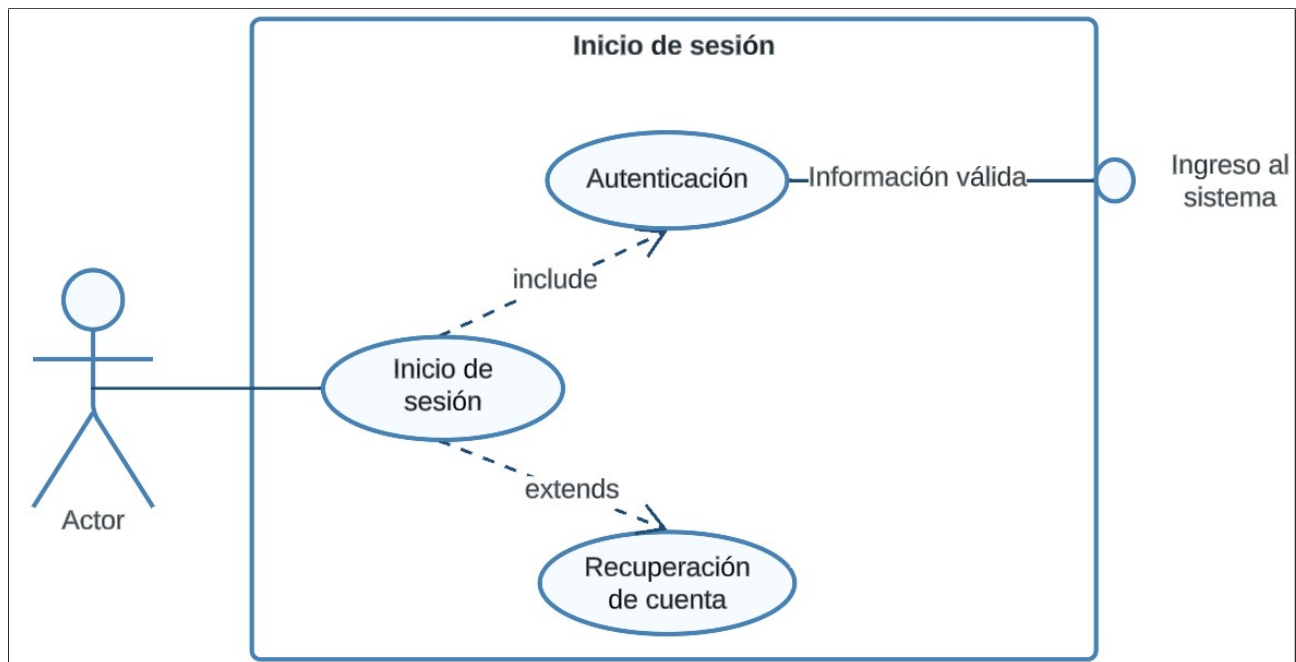


Figura 2: Diagrama de caso de uso: Inicio de sesión

4.2. Caso de uso: Realizar solicitud de examen

Caso de uso: Solicitud de examen de admisión

Actor: Usuario registrado

Precondiciones

1. El usuario ha ingresado al sistema.
2. El usuario no ha solicitado un examen de admisión en el año en curso.

Flujo principal

1. El usuario navega a la sección de "Examen de Admisión".
2. El sistema muestra distintas fechas/horas disponibles para realizar el examen.
3. El usuario selecciona una fecha/hora que le convenga.
4. El usuario hace clic en el botón "Realizar solicitud".
5. El sistema valida la disponibilidad y la elección del usuario.
6. Si la selección es válida y está disponible, el sistema genera un informe con los detalles de la solicitud y del examen.
7. El sistema muestra el informe al usuario.
8. El sistema envía la información del informe al correo del usuario.

Flujo alternativo

5a. Fecha/hora ya no disponible o inválida.

- El sistema muestra un mensaje de error indicando que la fecha/hora seleccionada ya no está disponible o es inválida.
- El usuario puede volver a intentar seleccionar otra fecha/hora.

5b. Usuario ya ha solicitado examen en el año en curso.

- El sistema muestra un mensaje de error indicando que el usuario ya ha solicitado un examen de admisión en el año actual.
- Se impide la continuación del proceso.

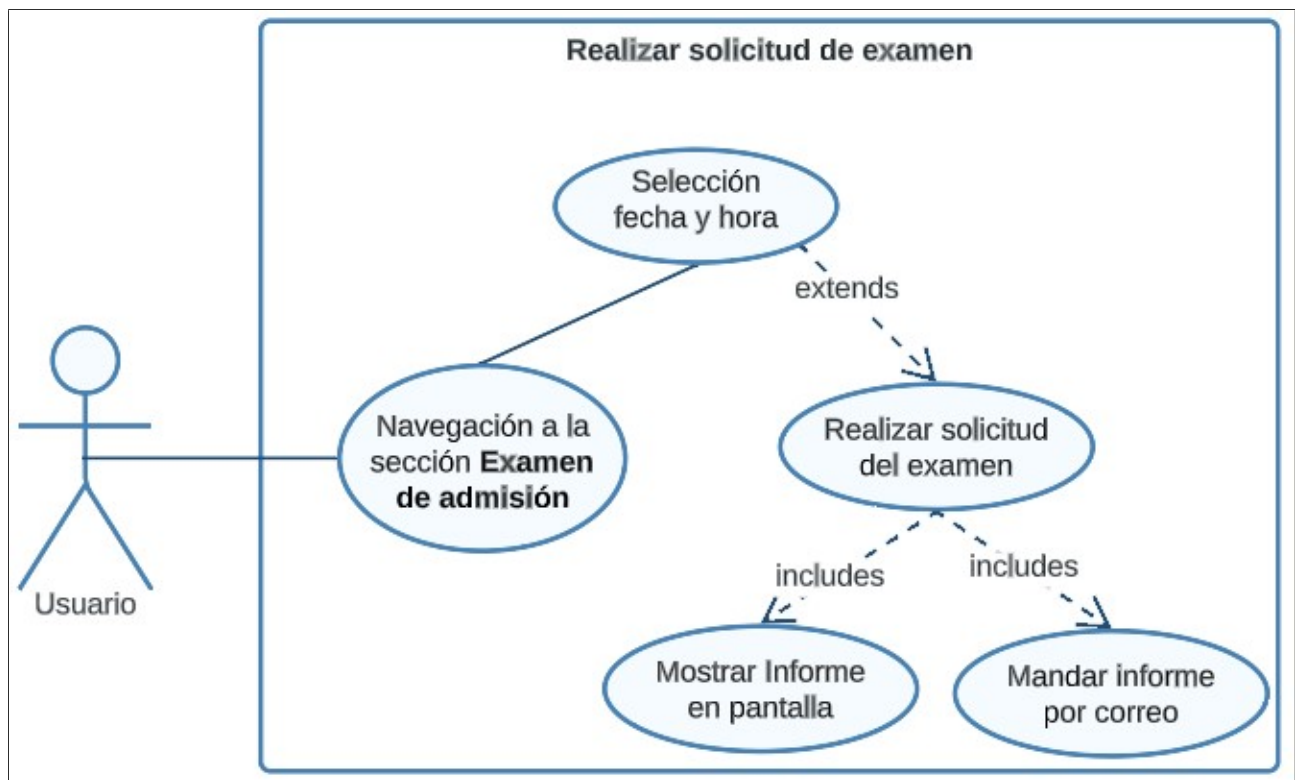


Figura 3: Diagrama de caso de uso: Realizar solicitud de examen

4.3. Caso de uso: Simular notas de exámenes

Caso de uso: Simulación de notas de examen

Actor: Administrador

Precondiciones

1. El administrador ha ingresado al sistema.
2. Hay solicitudes de examen para el período actual.

Flujo principal

1. El administrador navega a la sección de "Solicitudes de Examen".
2. El sistema muestra las solicitudes del período actual.
3. Opcionalmente, el administrador puede filtrar las solicitudes según la carrera.
4. El administrador selecciona las solicitudes a las cuales desea asignarles notas.
5. El administrador hace clic en el botón "Simular notas".
6. El sistema genera y asigna una nota aleatoria de admisión a las solicitudes seleccionadas.
7. Se muestra una confirmación de que las notas han sido asignadas.

Flujo alternativo

3a. No se realiza ningún filtrado.

- El administrador ve todas las solicitudes sin filtrar por carrera.

5a. Ninguna solicitud es seleccionada.

- El sistema muestra un mensaje de error indicando que se deben seleccionar solicitudes.

7a. Error al simular notas.

- El sistema muestra un mensaje de error indicando que no se pudieron asignar las notas.

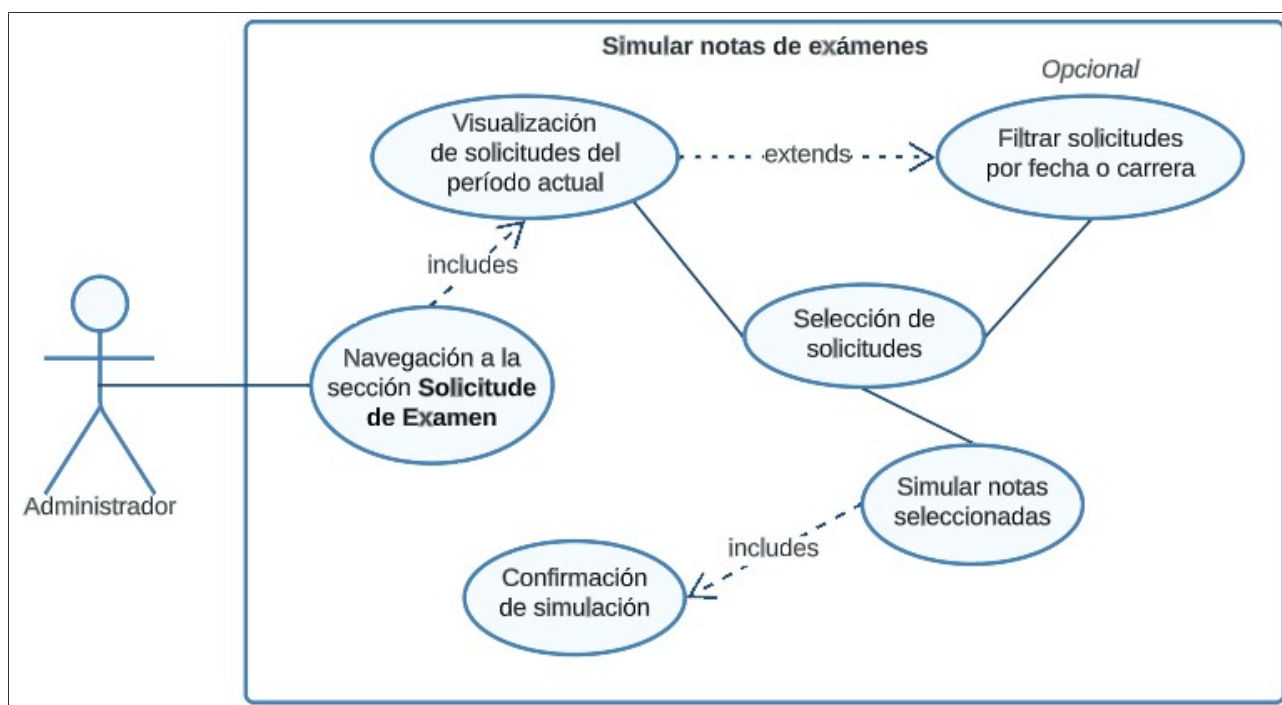


Figura 4: Diagrama de caso de uso: Simular notas de exámenes

4.4. Caso de uso: Asignar estados de solicitudes

Caso de uso: Asignación de Estado a Solicitudes

Actor: Administrador

Precondiciones

1. El administrador ha ingresado al sistema.
2. Existen solicitudes con notas asignadas.

Flujo principal

1. El administrador navega a la sección de "Solicitudes con Notas".
2. El sistema muestra todas las solicitudes que tienen notas asignadas.
3. Opcionalmente, el administrador puede filtrar las solicitudes.
4. Selecciona las solicitudes y les asigna un estado (admitido, en proceso, rechazado).
5. El sistema confirma la actualización del estado de las solicitudes seleccionadas.

Flujo alternativo

4a. Ninguna solicitud es seleccionada.

- El sistema muestra un mensaje indicando que se deben seleccionar solicitudes para actualizar el estado.

5a. Error al actualizar el estado.

- El sistema muestra un mensaje de error indicando que no se pudo actualizar el estado de las solicitudes.

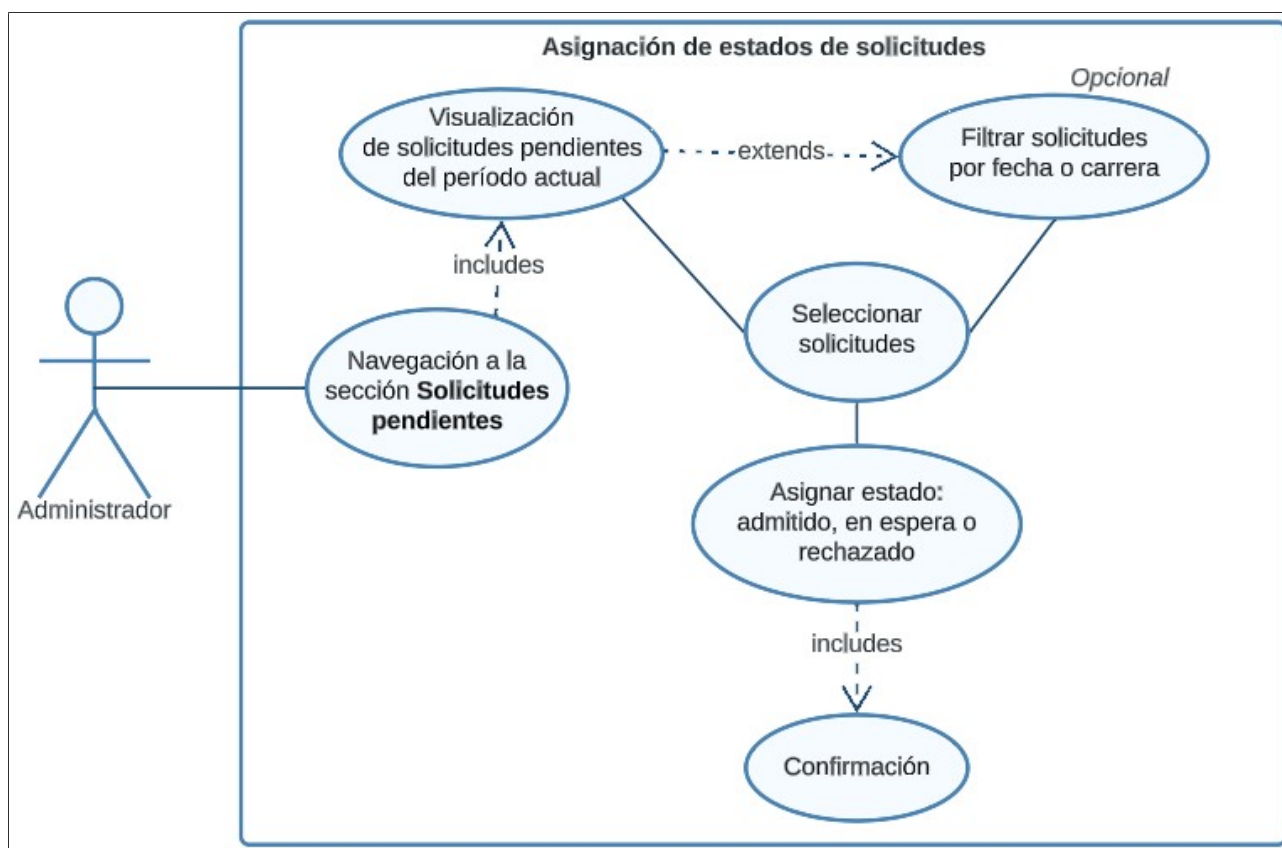


Figura 5: Diagrama de caso de uso: Asignar estado a solicitud

4.5. Caso de uso: Generar reportes

Caso de uso: Generación de Reportes

Actor: Administrador, Solicitante

Precondiciones

1. El administrador/solicitante ha ingresado al sistema.
2. Existen datos de exámenes de admisión por lo menos de un año.

Flujo principal

1. El administrador/solicitante navega a la sección de "Solicitudes pendientes".
2. El administrador selecciona un año de aplicación de examen de admisión.
3. El administrador selecciona un tipo de reporte.
4. El administrador solicita al sistema generar el reporte.
5. El sistema muestra una visualización preliminar del reporte.

Flujo alternativo

4a. Error al generar el reporte.

- El sistema muestra un mensaje de error indicando que no se pudo generar el reporte.

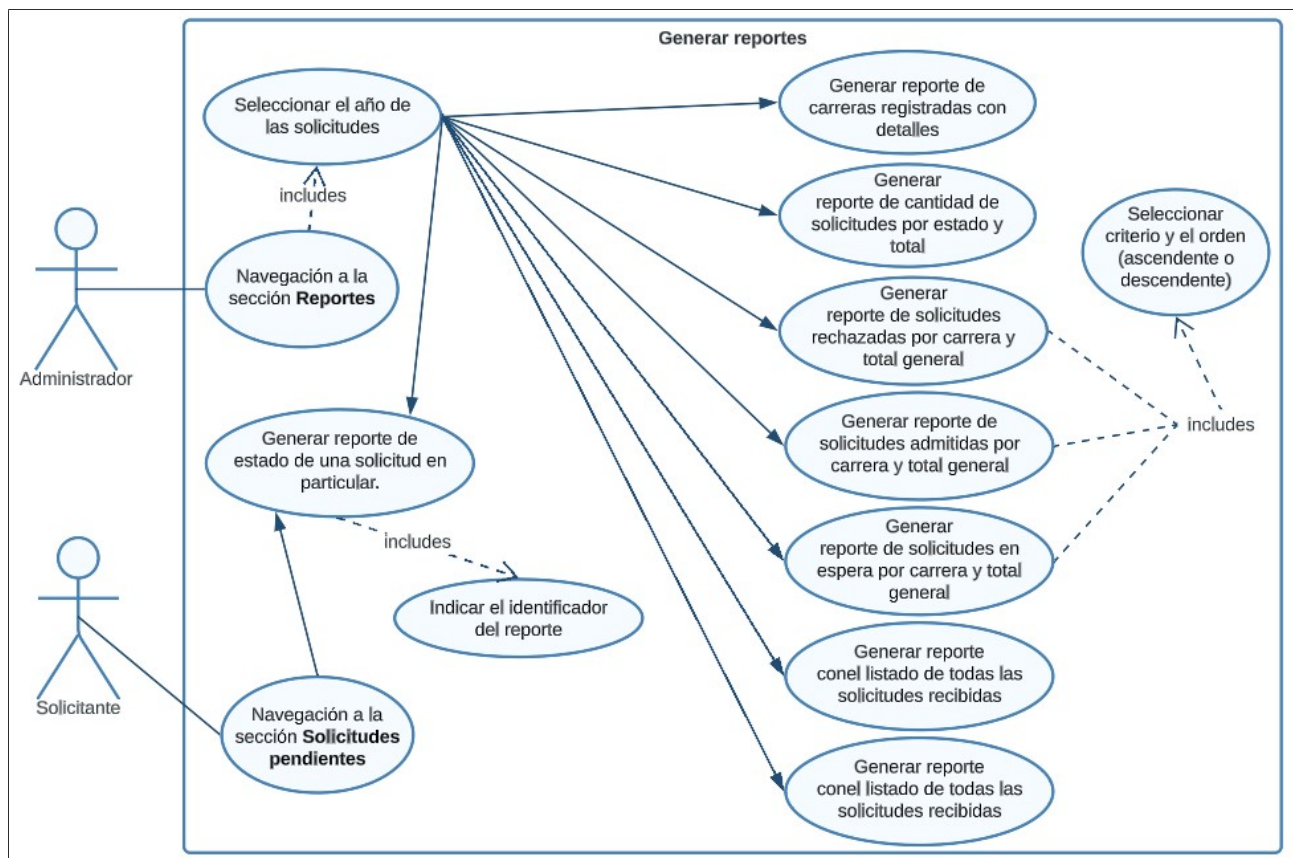


Figura 6: Diagrama de caso de uso: Generar reportes de solicitudes

4.6. Caso de uso: Gestionar cortes de carreras

Caso de uso: Gestión de cortes de carrera

Actor: Administrador

Precondiciones

1. El administrador ha ingresado al sistema.
2. Existen carreras registradas en el sistema.

Flujo principal

1. El administrador consulta la lista de carreras.
2. El administrador busca una carrera por nombre usando el buscador.
3. El administrador selecciona una carrera de la lista.
4. El administrador modifica la nota mínima de admisión, la cantidad máxima de admitidos por año o ambos.
5. El sistema valida la información ingresada.
6. El sistema solicita confirmación al administrador sobre los cambios realizados.
7. El administrador confirma y el sistema guarda los cambios.

Flujo alternativo

5a. Información ingresada no válida.

- El sistema muestra un mensaje de error.
- El administrador puede corregir la información y reintentar.

6a. El administrador no confirma los cambios.

- Se cancelan los cambios y se retorna al estado anterior.

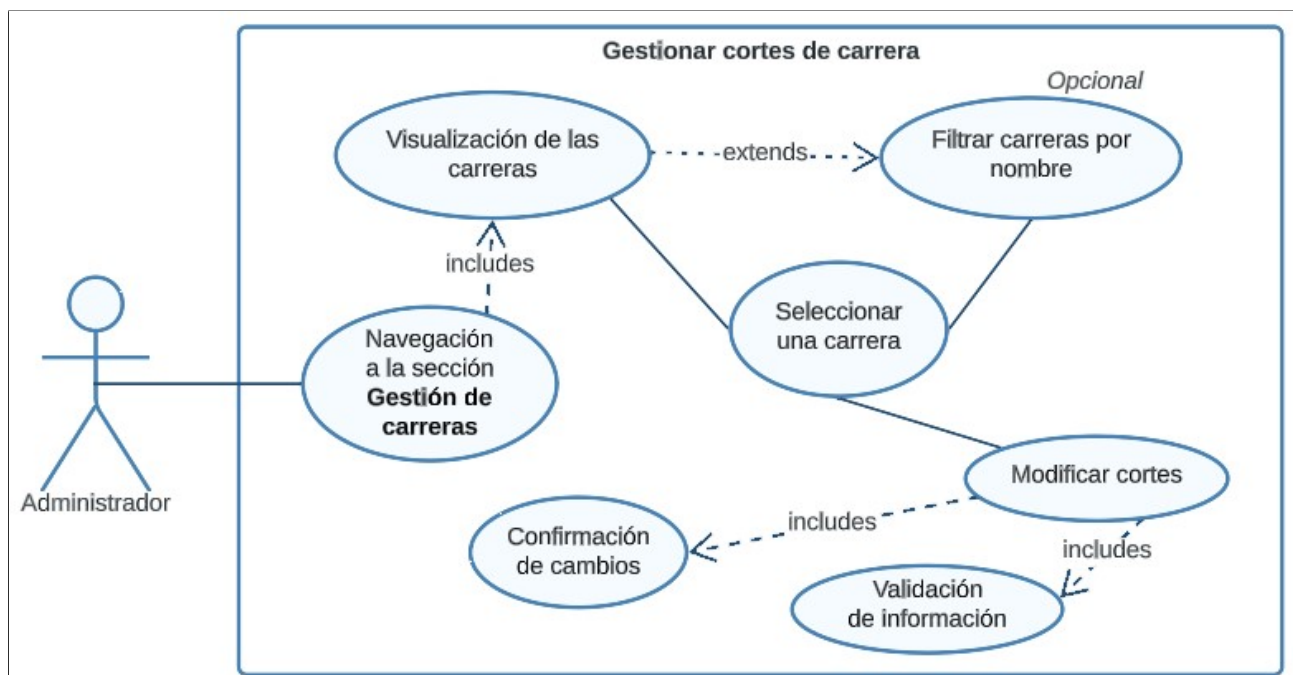


Figura 7: Diagrama de caso de uso: Gestión de cortes de carrera

5. Vista lógica

5.1. Modelo de negocios

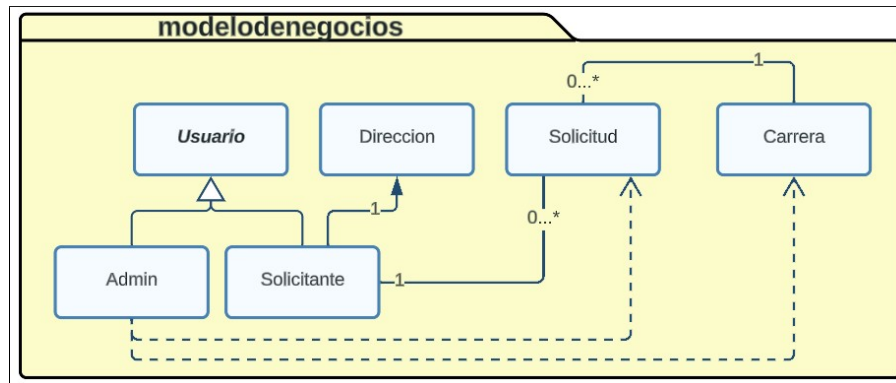
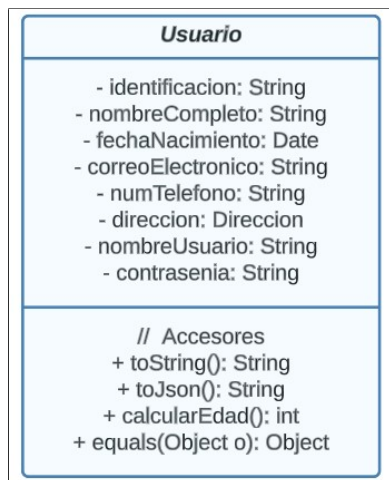


Figura 8: Modelo de negocios resumido

Primero, el sistema requiere de un proceso de registro e inicio de sesión. Se deben considerar tanto a los usuarios *administradores* y a los usuarios "normales". Para ello vamos a definir primero una clase abstracta denominada *Usuario*.

Clase Abstracta Usuario



Definimos una clase *Usuario* común con las características más básicas que tienen usuarios de esta vamos a derivar las clases *Admin* y *Solicitante*.

Figura 9: Clase Usuario

Clases Administrador, Solicitante y Dirección

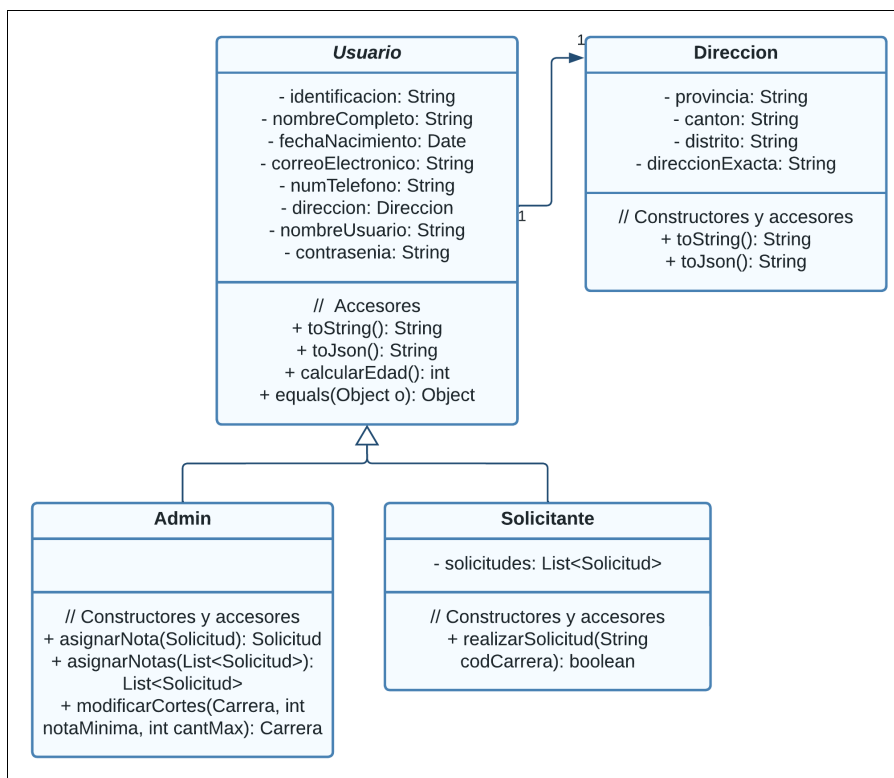


Figura 10: Clases Admin, Solicitante y Dirección

Se definió la clase *Dirección* por separado, y se definieron las clases de *Admin* y *Solicitante* en donde varían principalmente en los métodos que pueden realizar. Según la especificación el

- **Admin** puede asignar notas, nada más que esto se realizará de forma aleatoria; puede cambiar los cortes de una carrera, que implica que puede modificar la *nota mínima para ingresar* y la *capacidad máxima que se puede aceptar* para una carrera.
- **Solicitante** puede realizar la solicitud para realizar el examen de admisión indicando el código de la carrera.

La clase *Dirección* es muy trivial y no se va a detallar.

Lista de solicitudes

Los solicitantes tienen una lista de las solicitudes que han realizado en la plataforma. Eso se agregó considerando la propuesta de mantener la escalabilidad del sistema mencionada anteriormente.

Clases Solicitud y Carrera

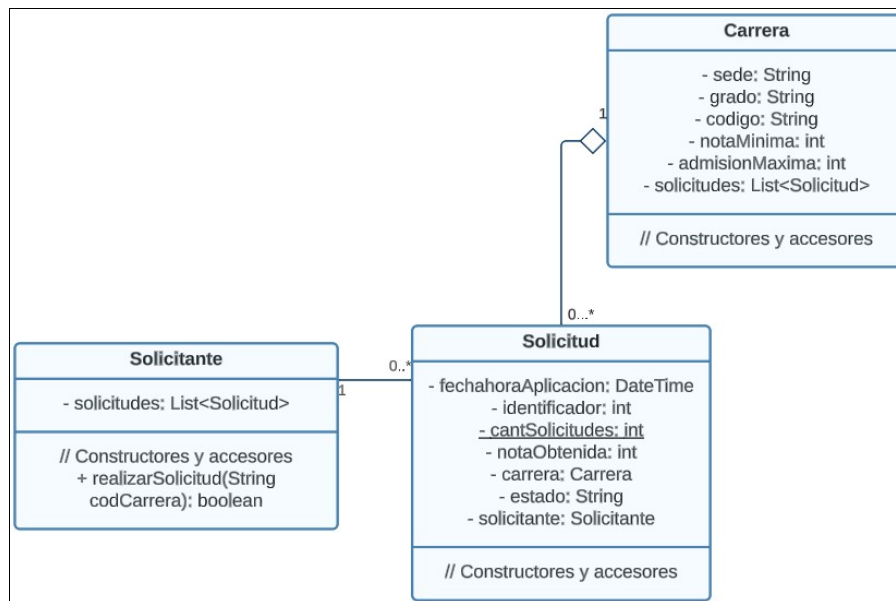


Figura 11: Clases Solicitud y Carrera

Por último, tenemos las clases de Solicitud y Carrera. No tienen ninguna funcionalidad en específico, el administrador se encargará de acceder a las instancias de estos objetos.

Justificación

Un solicitante puede tener varias solicitudes y además, una carrera estará agregada por varias solicitudes. Estas relaciones facilitarán la consulta para los servicios solicitudes del sistema.

El sistema completo con más detalle se visualizaría en la siguiente figura.

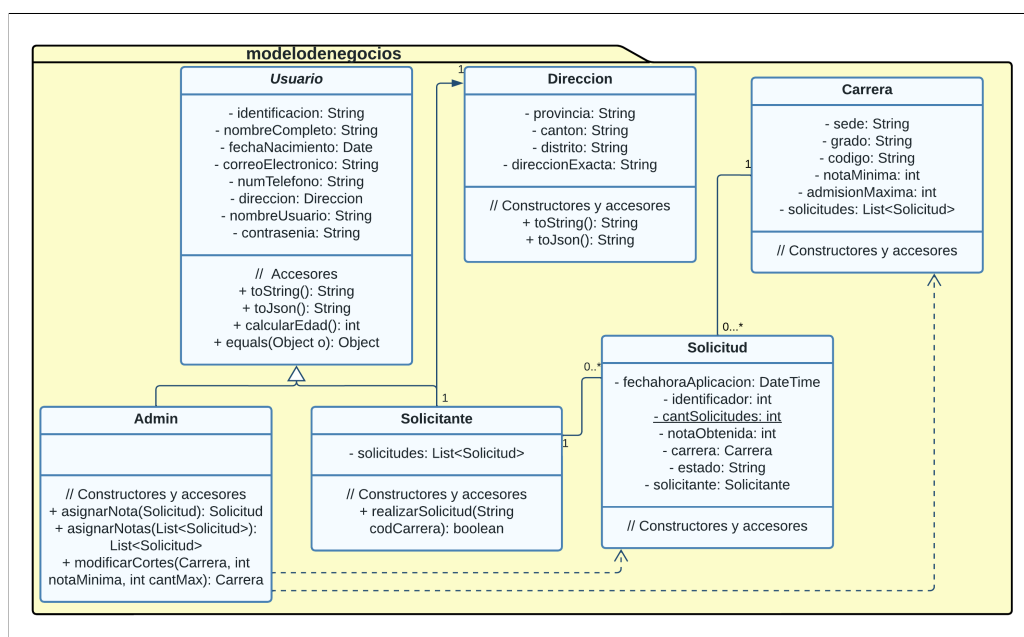


Figura 12: Modelo de negocios detallado

5.2. Controladores y Vistas

Como se mencionó anteriormente, la arquitectura del sistema será utilizando el patrón *modelo-vista-controlador* (MVC). Además, se utilizará el paquete de *Data Access Object* (DAO) para la estructuración de los datos obtenidos desde la base de datos y moldearlos apropiadamente con nuestro modelo de negocios. La organización de la lógica del sistema se puede representar con el siguiente diagrama:

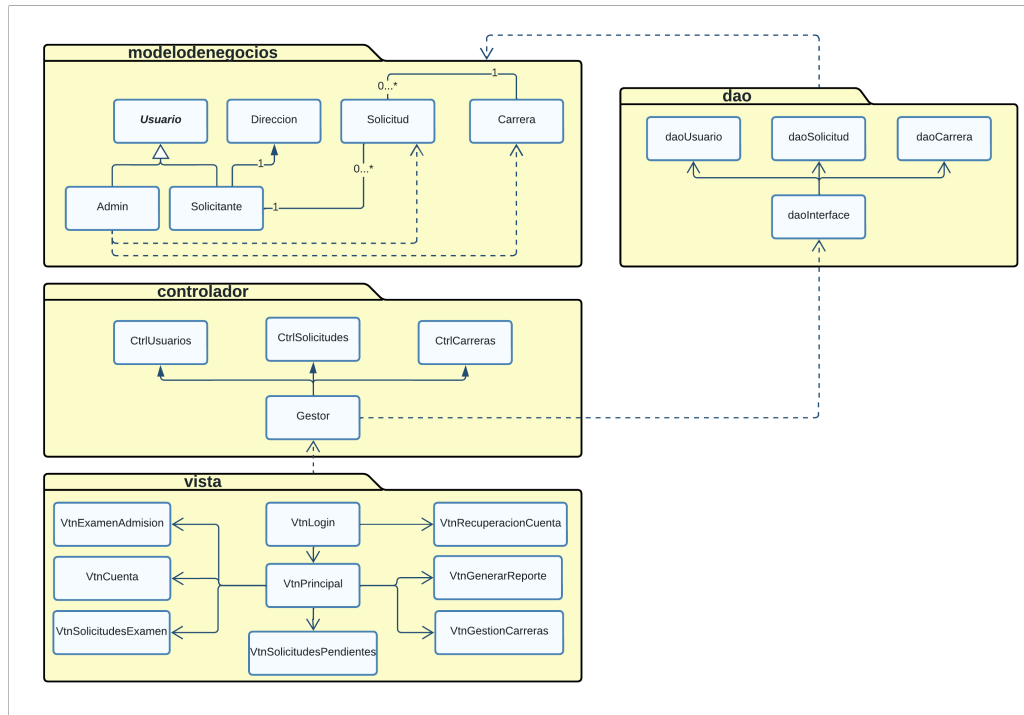


Figura 13: Arquitectura resumida

Los **controladores** se encargarán del apartado lógico de manipulación de datos. Recordando que:

- Un administrador puede modificar información de las carreras.
- Un administrador puede “simular” notas de exámenes (actualizando las solicitudes).
- Un administrador puede cambiar el estado de una solicitud como: **en proceso**, **rechazada** o **admitida**.
- Los usuarios pueden realizar solicitudes de exámenes de admisión una vez por año.

5.3. Propuestas de vistas

A continuación se presenta un modelo preliminar de las vistas en modo *Solicitante*. La ilustrar la idea principal del modelo propuesto para los solicitantes, las vistas del modo *administrador* no serán incluidas.

Ventana Login

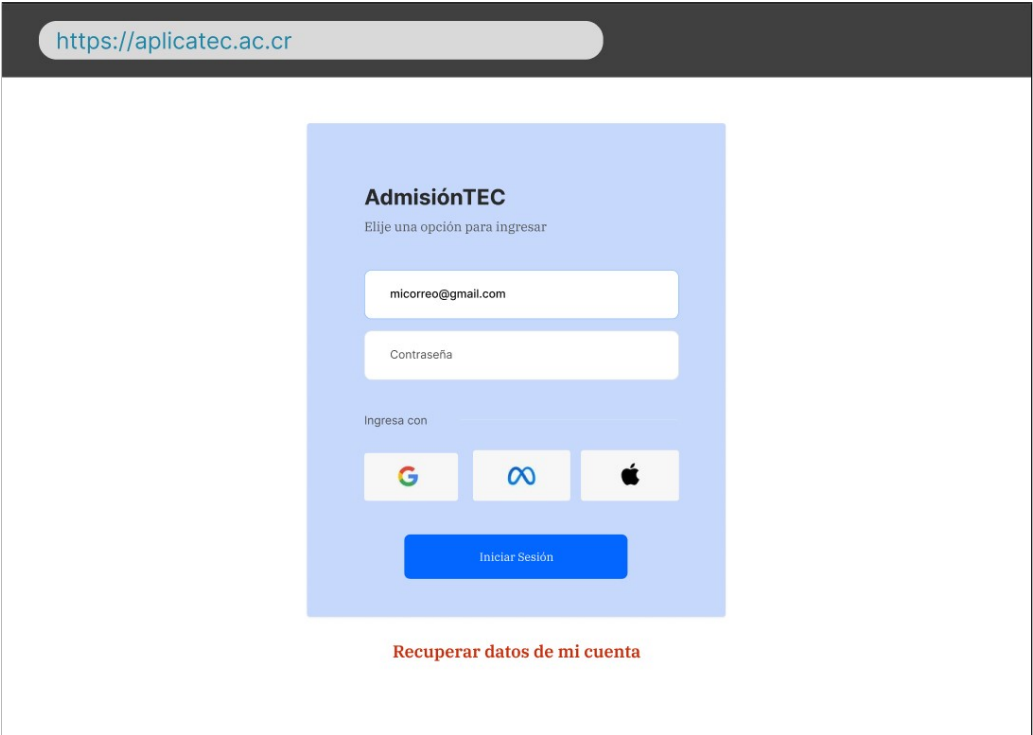


Figura 14: Ventana Login

Ventana principal



Figura 15: Ventana principal

Ventana para solicitar hacer el examen

The screenshot shows a web interface for applying for an exam. At the top, there is a URL bar with the address `https://aplicatec.ac.cr/examenes`. Below this, there are two dropdown menus: 'Sede' (Location) with 'Limón' selected, and 'Carrera' (Career) with 'Ing. En Computación' selected. Below the filters, there are three exam slots, each represented by a blue rounded rectangle. Each slot contains the text: 'Tecnológico de Costa Rica, sede de Limón', 'Dirección exacta de la institución', and 'Día 16 de Setiembre, [time]'. To the right of each slot is a blue button labeled 'Aplicar' (Apply).

Sede	Carrera
Limón	Ing. En Computación

Examen	Acción
Tecnológico de Costa Rica, sede de Limón Dirección exacta de la institución Día 16 de Setiembre, 2pm	Aplicar
Tecnológico de Costa Rica, sede de Limón Dirección exacta de la institución Día 16 de Setiembre, 5pm	Aplicar
Tecnológico de Costa Rica, sede de Limón Dirección exacta de la institución Día 16 de Setiembre, 8pm	Aplicar

Figura 16: Ventana de citas para el examen de admisión

6. Vista de deployment

A continuación se ilustrará una descripción breve de la configuración del deployment del sistema. En esta aplicación se asume que hay una base de datos externa que tiene información correspondiente a las carreras, en nuestro modelo definimos una base de datos que albergará la información de los solicitantes y sus respectivas solicitudes.

Además, se integrará un módulo para la gestión de correos electrónicos considerando que el sistema notificará a los usuarios cuando realicen las solicitudes y cuando se les notifique su nota del examen de admisión al cuál aplicaron. Junto con un módulo de validación de cuentas de forma segura, considerando que se maneja información sensible en este sistema.

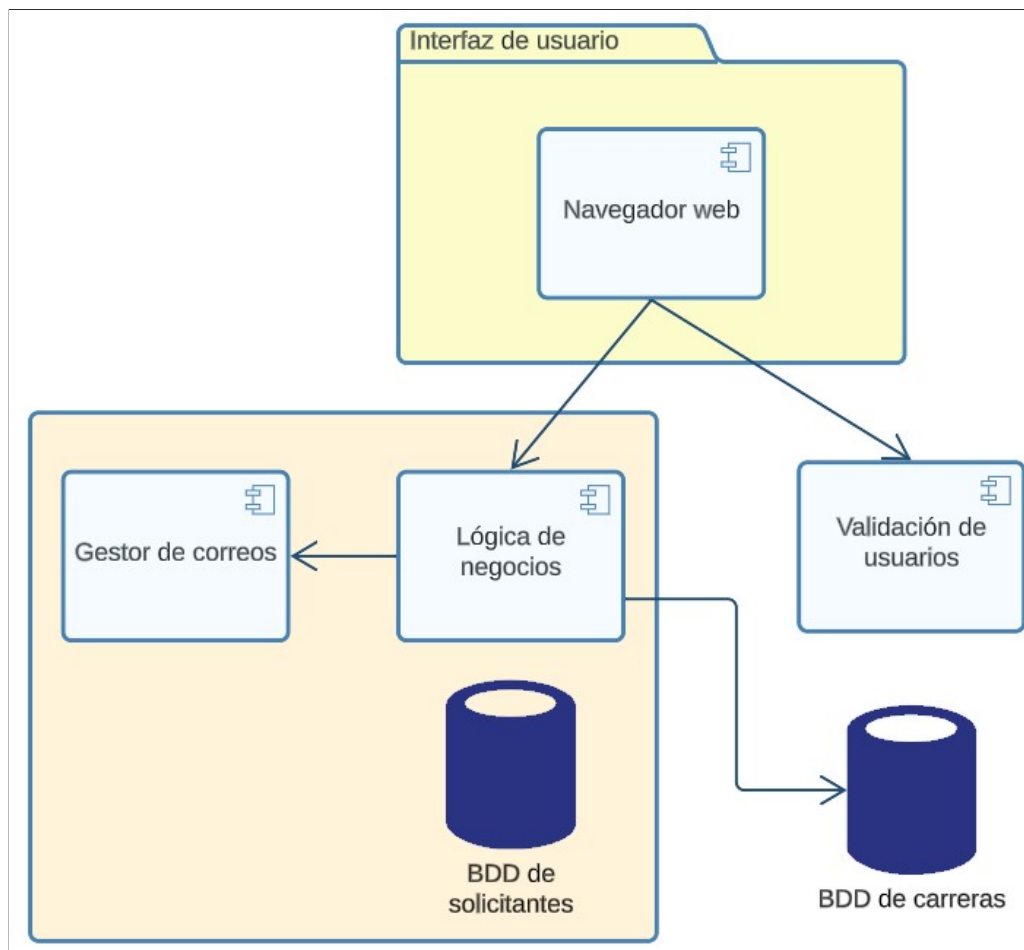


Figura 17: Resumen del deployment de la aplicación

7. Calidad

A continuación se mencionan algunos atributos importantes y puntuales referentes a la calidad del sistema:

- El sistema deberá ser capaz de manejar hasta 10000 solicitudes de forma simultánea sin degradar el rendimiento. Para efectos de costo, se consideran nada más los períodos clave (durante los procesos de admisión del año).
- El sistema deberá ser compatible con las bases de datos MySQL (se asume que la base de datos de las carreras este motor de base de datos).
- El sistema debe ser robusto, la seguridad es importante para evitar accesos no autorizados y resguardar la información sensible de la población. Además de evitar la alteración indebida de datos.
- El sistema debe estar disponible 24/7. No debe tener más de un 5% de inactividad.
- La interfaz debe ser fácil de usar. Además, se contará con un sistema guía para las personas que estén ingresando a la aplicación por primera vez.
- El soporte deberá estar integrado al sistema, abierto y disponible para las consultas de los usuarios registrados.

8. Tamaño y rendimiento

Los principales requisitos de tamaño y tiempo se definen:

- El sistema deberá proporcionar acceso a la base de datos con no más de 8 segundos de latencia.
- El sistema deberá ser capaz de completar el 85% de todas las transacciones dentro de 1.5 minutos.
- La parte del cliente requerirá menos de 25 MB de espacio en disco y 40MB de RAM.