

Actividad 6 - Entrega final del proyecto

David Bernal

Ingenieria de Software, Universidad Iberoamericana

Proyecto de software

TATIANA CABRERA

Diciembre 12, 2025

Tabla de contenido

1. Introducción	3
2. Justificación	3
3. Contextualización.....	3
4. Planteamiento del problema	4
5. Alcance	4
6. Objetivos	5
7. Estructura de desglose EDT	6
8. Metodología ágil	7
9. Presupuesto	7
10. Levantamiento de información	8
11. Conclusión levantamiento de información	10
12. Mapa de Stakeholders	10
13. Matriz de riesgos	11
14. Diagrama de flujo de solución	11
15. Requisitos funcionales no funcionales	12
16. Historias de Usuario	13
17. Enlace de Repositorio	16
18. Conclusiones	16
19. Referencias	16
20. Casos de uso.....	17
21. Diagrama de secuencia.....	22
22. Diagrama de clases.....	23

23. Diagrama de componentes.....	24
24. Diagrama de despliegue.....	24
25. Modelo de base de datos.....	25
26. Prototipo de baja fidelidad.....	26
27. Prototipo de alta fidelidad.....	26
28. UserMap.....	27
29. UserFlow.....	27
30. Pruebas de usabilidad.....	28
31. Patrones de arquitectura.....	30
32. Diagrama de arquitectura.....	31
33. Estructura del proyecto.....	31
34. Elevator pitch.....	31
35. Repositorio.....	31
36. Conclusiones.....	32
37. Referencias.....	33

- **Introducción**

El desarrollo de aplicativo de software requiere generar una fase de análisis y planeación, estas fases deben ser bien documentadas para poder tener un contexto, objetivos y unas definiciones en cuanto a lo que se espera de la solución y como está va a ser desarrollada, de la calidad del documento depende que se evite caer en dudas o falsas interpretaciones que posteriormente va a generar retrabajos aumentando costos y tiempos de entrega, entre otros. Este trabajo tiene apartes y está basado en trabajos anteriores realizados por mi autoría, pero modificados de acuerdo a los lineamientos y objetivos de la materia proyecto de grado.

- **Justificación**

Es requerido la planificación del desarrollo de software como parte importante de la solución a un problema, en el ciclo de vida del desarrollo del software buscamos estructurar el proyecto y como parte de esa estructuración se espera definir todas las partes que la componen, la primera fase estará enfocada al análisis, donde se trabajara alrededor del problema, como se abordara y que se buscara obtener, posteriormente se analizara la información recabada y el planteamiento de la solución, para la definición de las funcionalidades.

La solución estará basada en entregar un software capaz de gestionar la autorización y el registro de visitantes y que tanto residentes como visitantes puedan generar un ingreso fácil y simple.

FASE DE PLANIFICACIÓN DEL CICLO DE VIDA DEL DESARROLLO DE SOFTWARE

- **Contextualización de la necesidad, (Bernal 2025a)**

El edificio de oficinas actualmente no cuenta con un control de acceso de visitantes y tampoco el registro de los mismos, esto lleva a tener problemas de control de acceso, verificación de la salida de los visitantes y riesgos de seguridad en edificios empresariales, se requiere un sistema digital que automatice el registro y la autorización de los visitantes que sea eficiente y no genere traumas en el ingreso al edificio.

- **Planteamiento del problema, (Bernal 2025a)**

¿Cómo desarrollar un software que automatice el registro de visitantes y su autorización para la gestión del acceso al edificio empresarial, garantizando seguridad y eficiencia en la operación?

- **Alcance del proyecto. (Bernal 2025a)**

Aunque el proyecto completo busca tener un sistema de control de acceso a propietarios y visitantes, que estos puedan a través de torniquetes, hacer el ingreso y salida del edificio, la parte administrativa busca generar informes y hacer búsqueda de visitantes, solo se va a desarrollar el módulo de registro y autorización de visitantes para el desarrollo de la actividad.

- **Modulos**

- Módulo de registro de visitantes

- Desarrollar funciones en el servidor local para ser puente entre la base de datos e interfaz.
 - Desarrollar funciones en la interfaz de usuario
 - Realizar pruebas de conectividad con base de datos
 - Realizar pruebas de interfaz
 - Realizar pruebas de integración

- Módulo de autorización de visitantes

- Implementar base de datos relacional para almacenar los datos
 - Desarrollar funciones en el servidor local para ser puente entre la base de datos e interfaz.
 - Desarrollar funciones en el interfaz de usuario
 - Realizar pruebas de conectividad con base de datos
 - Realizar pruebas de interfaz
 - Realizar pruebas de integración

- **Restricciones**

- Aunque la solicitud del cliente es que funcione localmente para evitar la dependencia de internet, por los requerimientos del curso se implementara de forma web para su entrega.
 - El tiempo de entrega no debe ser superior a 2 meses

- **Criterios de aceptación**

Autorización de visitantes

- Crear autorización a visitante en menos de 1 minuto
- Autorización de visitantes en un solo paso
- Validar campos vacíos y que coincida el tipo de campo
- Conexión a base de datos para almacenar registros
- Generar un mensaje de estado de registro

Registro de visitantes

- Encontrar autorización de visitantes en menos de 3 segundos
- Registro de visitante en un solo paso
- Validar campos vacíos y que coincida el tipo de campo
- Conexión a base de datos para actualizar registros
- Generar un mensaje de estado de registro

- **Objetivos. (general y específicos)**

Objetivo General

Desarrollar e implementar un software que automatice el registro de visitantes y su autorización para la gestión del acceso al edificio empresarial

Objetivos Específicos

- Recolectar y analizar los requisitos de los diferentes usuarios del software
- Implementar la base de datos y los datos que requieren ser almacenados
- Desarrollar la interfaz de usuario para usuarios de seguridad y residentes
- Ejecutar pruebas al software

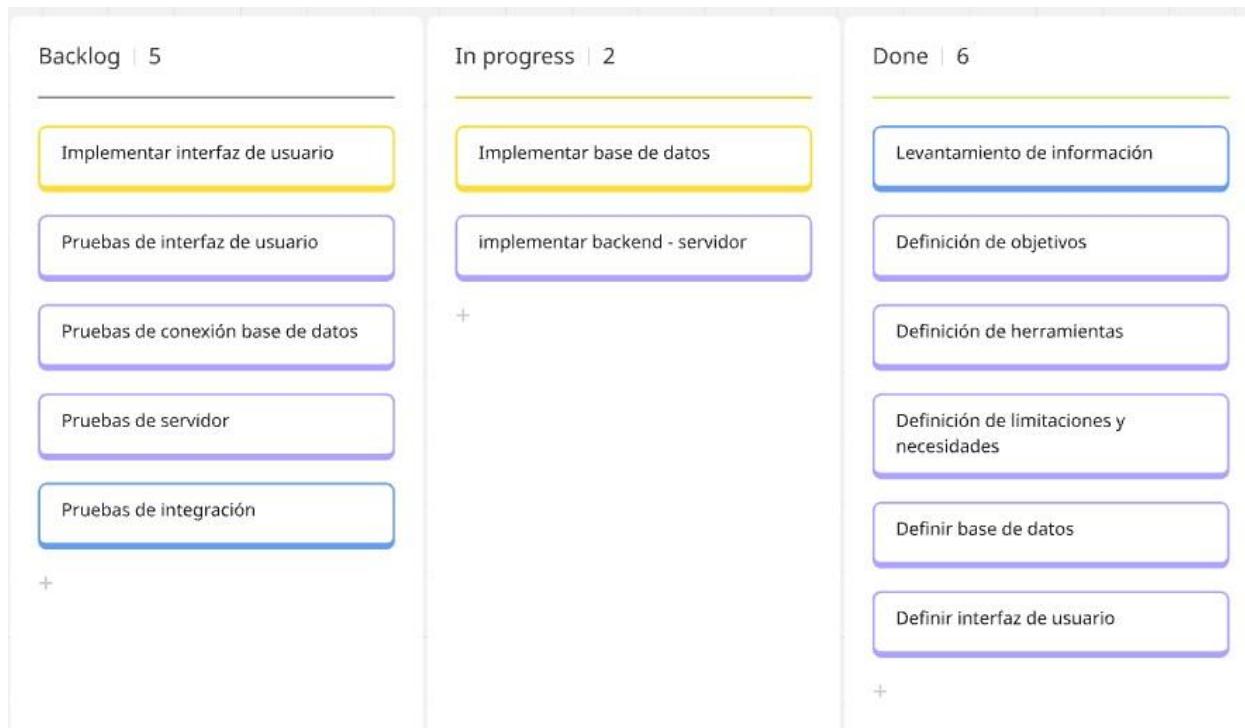
- **Estructura del Desglose. (EDT), (Bernal 2025b)**

Enlace: <https://mm.tt/map/3434817579?t=NMwaWLFDrY>



- **Metodología ágil seleccionada (Enlace de Tablero)**

Enlace: https://miro.com/app/board/uXjVIHU2zBw=?share_link_id=



- **Presupuesto**

Costos desarrollo de Software Control de Acceso		
Item	Costo x 2 meses	
Gerente de proyecto	\$	2,000
Desarrollador fullstack	\$	3,000
Diseñador grafico	\$	1,000
QA	\$	1,300
Servidor	\$	600
Capacitación a usuarios	\$	150
Sub-Total - USD	\$	8,050
Contingencia 10% -USD	\$	805
Total - USD	\$	8,855

FASE DE ANALISIS DEL CICLO DE VIDA DEL DESARROLLO DE SOFTWARE

- **Levantamiento de información (Bernal, 2025a)**

-Diseño de instrumento

Para esta etapa se va a realizar un proceso de entrevista semiestructurada a diferentes actores al personal de administración y seguridad del edificio, la población la comprende el administrador del edificio, auxiliar de administración y 2 guardas, el objetivo es lograr entender las problemáticas y necesidades del edificio.

Preguntas a administración

- ¿Llevan registro de visitantes?
- ¿Como autorizan el ingreso de un visitante?
- ¿Qué problemas presentan actualmente?
- ¿Qué funcionalidad requieren que tenga un control de visitantes?
- ¿Qué información esencial requiere para el ingreso de un visitante?
- ¿Qué nivel de conocimiento tecnológico tienen los guardas?

Preguntas a guardas

- ¿Que se solicita a los visitantes que ingresan al edificio?
- ¿Como se verifica el acceso de un visitante?
- ¿Qué datos se piden a un visitante?
- ¿Se transmite algún tipo de información de los visitantes entre el cambio del turno?

- Aplicación del instrumento

Se diseñaron preguntas abiertas enfocadas al administrador y su auxiliar y para los guardas, se explicó el propósito de la entrevista y la necesidad de hondar en el detalle para entender correctamente el problema y asegurar las necesidades y se realizó de forma virtual a través de una llamada.

-Informe ejecutivo

Se realizó la entrevista el 2 de abril de 2025

Se entrevistaron 4 personas

Entrevista a administración – Tabla 1

Pregunta	Respuesta
¿Llevan registro de visitantes?	No, Porque no había sido necesario llevar el registro de los visitantes
¿Como autorizan el ingreso de un visitante?	No, se tiene un sistema de comunicación con los propietarios
¿Qué problemas presentan actualmente?	No, se sabe cuántos visitantes hay y en qué lugar están o quien les autorizo el ingreso
¿Qué funcionalidad requieren que tenga un control de visitantes?	Se requiere que tenga un informe detallado de los visitantes que ingresan, se requiere el registro de los datos básicos y su fotografía
¿Qué información esencial requiere para el ingreso de un visitante?	Los datos básicos, nombre, apellidos, documento de identificación, oficina de destino y quien autorizo el ingreso del visitante

Entrevista a guardas – Tabla 2

Pregunta	Respuesta guarda 1	Respuesta guarda 2
¿Que se solicita a los visitantes que ingresan al edificio?	Mostrar el documento y decir a que oficina se dirige	Mostrar el documento y quien autorizo el ingreso
¿Qué datos se piden a un visitante?	Ninguno	Si no lo conozco le pido el nombre
¿Como se verifica el acceso de un visitante?	No se verifica, solo se da acceso si se que el propietario de la oficina se encuentra	No se puede verificar porque no hay un sistema para hacerlos
¿Se transmite algún tipo de información de los visitantes entre el cambio del turno?	Solo si hay alguna novedad	Cuando pasa algo relevante o cuando están haciendo algún trabajo o mantenimiento en alguna oficina

-Análisis y conclusiones de los resultados

De las entrevistas, se puede identificar que no tienen algún método de registro de visitantes, que entre las partes no hay un sistema de comunicación y que, aunque quieren un control de visitantes no tienen una idea clara de cómo lograrlo, aunque mencionan la necesidad del sistema como parte de un requerimiento de la aseguradora del edificio.

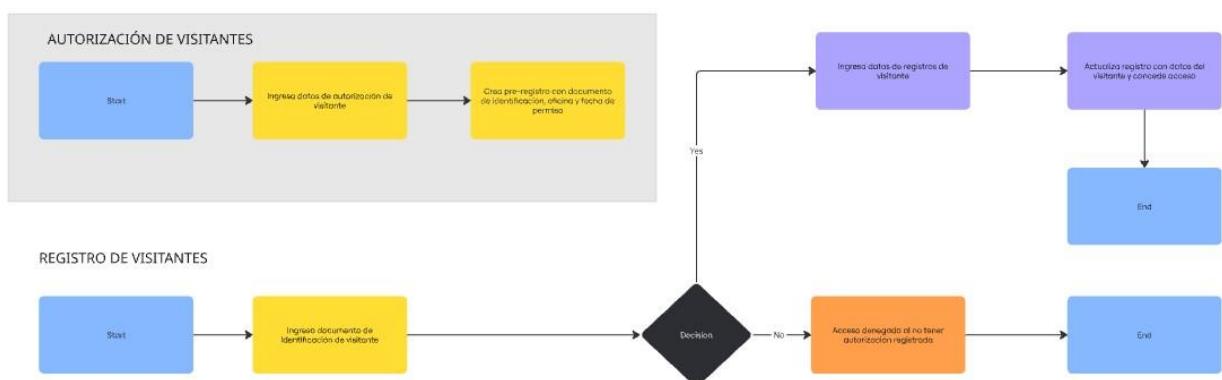
- **Mapa de Stakeholders y clasificación de los mismos**



- **Matriz de riesgos.**

Riesgo	Probabilidad	Impacto	Mitigación
Resistencia al cambio en guardas	Media	Alto	Capacitaciones, Manuales simples, acompañamiento
Resistencia al cambio en residentes	Media	Medio	Capacitaciones, Manuales simples, acompañamiento
Caída del servidor en horas pico	Baja	Alto	Redundancia, copias de seguridad, reinicio automático
Identificación errónea en la autorización	Media	Medio	Validación de campos, Verificación por llamada
Datos erróneos en el proceso de registro	Media	Medio	Validación automática de campos, verificación

- **Diagrama de flujo de solución**



- **Definición de RQF-RQNF manejar un formato para la documentación. (Bernal 2025a)**

-Requisitos funcionales

1. Autorizar visitante

-El sistema debe permitir registrar la autorización de un visitante.

-El sistema requiere para registrar la autorización los siguientes datos, oficina de destino, nombre de quien autoriza, contraseña de autorización, fecha de ingreso y número de identificación del visitante

2. Registrar Visitante

- El sistema debe permitir buscar un visitante que haya sido autorizado y si es encontrado, debe entregar los datos de la autorización.

- El sistema debe capturar el nombre y apellido del visitante

- Requisitos no Funcionales

1. Disponibilidad

- El sistema debe estar disponible 24/7

- El sistema no puede depender de servicio de internet

1. Seguridad

- El sistema debe contar con un sistema de roles y con la autenticación correspondiente para cada tipo de usuario

- La información sensible debe guardarse con métodos de seguridad acordes al tipo de información.

- El sistema no podrá ser accedido por usuarios o administradores fuera de la red del edificio.

- El sistema deberá contar con un sistema de confirmación dentro del edificio para realizar labores de mantenimiento.

3. Usabilidad

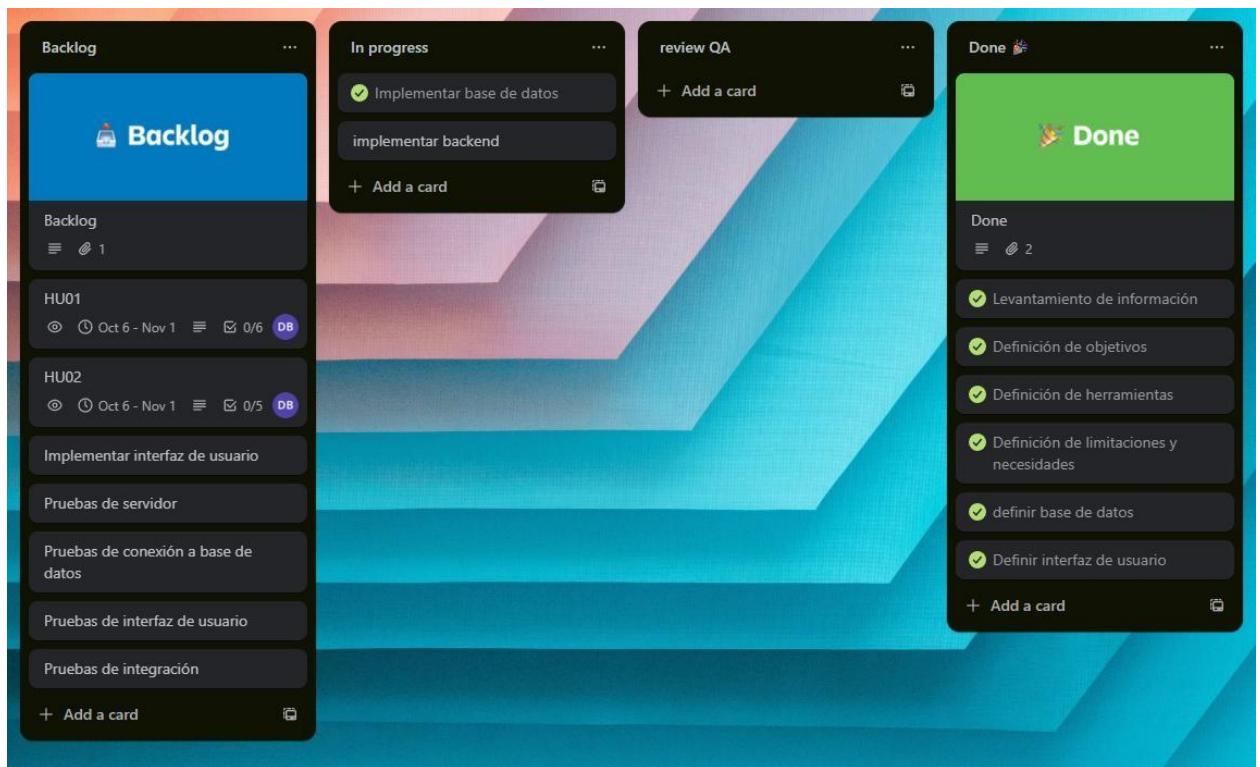
- El sistema debe ser de fácil uso e intuitivo y debe requerir la menor cantidad de pasos posibles para realizar los procesos.

- El sistema debe correr sobre un navegador web

- El sistema debe contar con mensajes de alertas y errores claros y explicativos

- Historias de Usuario en el tablero de la metodología Ágil seleccionada. (Bernal 2025a)

Enlace: <https://trello.com/b/g54nb43C>



- HU01, Como propietario de oficina, quiero autorizar un visitante rápidamente, para un ingreso fácil y rápido a la oficina.

The screenshot shows a user story card for 'HU01' on a dark-themed interface. At the top left is a green checkmark icon followed by the identifier 'HU01'. To the right is a '+ Add' button. Below the identifier are two circular icons: a purple one labeled 'DB' and a grey one with a plus sign. Underneath these is a section titled 'Description' with a three-line menu icon to its left. To the right of 'Description' is an 'Edit' button. The main body of the card contains the user story text:

Como propietario de oficina,
quiero autorizar un visitante rápidamente,
para un ingreso fácil y rápido a la oficina.

Below the story text is a section titled 'Puntos de esfuerzo: 8' (Effort Points: 8). Underneath this is a section titled 'Criterios de aceptación' (Acceptance Criteria) with a checked checkbox icon. To the right of this section is a 'Delete' button. A progress bar below the acceptance criteria shows 0% completion. The acceptance criteria list includes the following items, each with an unchecked checkbox:

- Crear un formulario que contenga los campos, oficina, persona a quien visita, contraseña de autorización, fecha de ingreso del visitante y documento del visitante
- Validar campos vacíos y que coincida el tipo de campo
- Crear la funcionalidad de registrar una autorización
- Conectar la base de datos para almacenar los datos
- Generar un mensaje de estado de la autorización al finalizar
- Registro de autorización menor a 1 minuto

- **HU02, Como guardia de seguridad, quiero registrar un visitante autorizado rápidamente para que el visitante pueda dirigirse a la oficina de destino.**

The screenshot shows a user story card for "HU02". At the top left is a radio button followed by the identifier "HU02". To the right is a "+ Add" button. Below this is a header section with a "DB" icon, a "+" button, and the date range "Oct 6 - Nov 1, 7:04 PM". To the right of the date is a "Description" section with an "Edit" button. The description text is as follows:

Como guardia de seguridad,
quiero registrar un visitante autorizado rápidamente
para que el visitante pueda dirigirse a la oficina de destino.

Below the description is a section titled "Puntos de esfuerzo: 8". Underneath this is a section titled "Criterios de aceptación" with a checked checkbox and a "Delete" button. A progress bar indicates 0%. The acceptance criteria list contains the following items:

- Crear un formulario que contenga los campos ,documento del visitante, persona a quien visita, oficina a la que se dirige, fecha de ingreso del visitante, nombre y apellido del visitante
- Generar la validación de campos vacíos y que coincida el tipo de campo
- Crear la funcionalidad de registrar un visitante
- Conectar la base de datos para consultar y actualizar los datos
- Generar un mensaje de estado del registro al finalizar

- Repositorio

Enlace: <https://github.com/davidanbernal/ProyectoDeSoftware>

Conclusiones

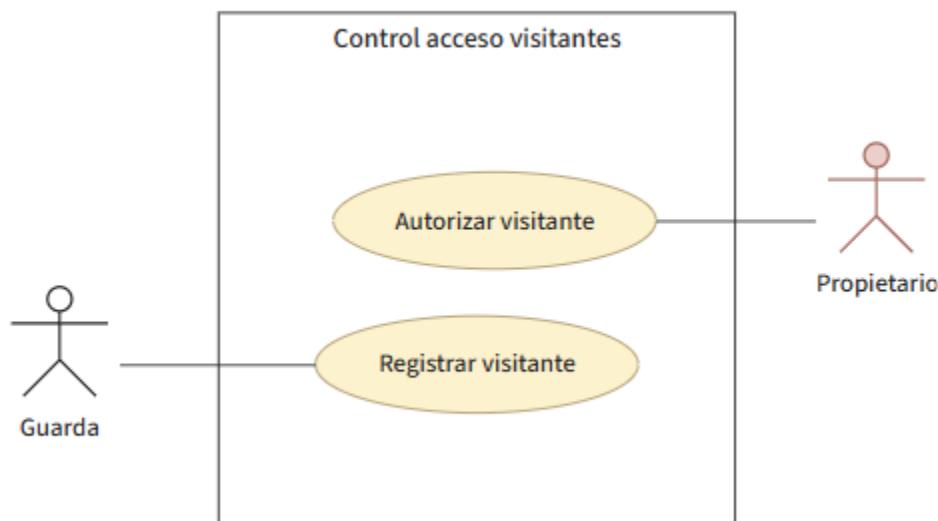
- La documentación es la columna vertebral de las buenas prácticas en el desarrollo de sistemas de software, entre que la documentación sea más detallada mayor éxito va a tener el desarrollo de la misma.
- El tener objetivos claros y una hoja de ruta, permite que todos los involucrados en el proyecto identifique su rol y marque las tareas correspondientes para el desarrollo de sus actividades y delimitando el sistema.

Referencias

- Bernal, D. (2025a). Actividad 1,2,3, Análisis y Diseños de sistemas [Manuscrito no publicado]. Universidad Iberoamericana.
- Bernal, D. (2025b). Actividad 3, 4, 5, 6, Gerencia y control de proyectos informáticos [Manuscrito no publicado]. Universidad Iberoamericana.

SEGUNDA ENTREGA

- **Modelos de comportamiento y estructura:**
 - Casos de uso documentados



1. IDENTIFICACIÓN DE CASO DE USO			
1.1 Id Caso	CU-01	1.2 Nombre	Autorizar visitante
2. HISTORICO DE CASO DE USO			
2.1 Autor	David Bernal	2.3 Última Actualización	
2.2 Fecha Creación	15/11/2025	2.4 Actualizado por	2.5 Versión
2.4 Actualizado por			1.0
3. DEFINICIÓN DE UN CASO DE USO			
3.1 DESCRIPCIÓN			
Permite al propietario de oficina crear una autorización de ingreso para un visitante. El propietario llena un formulario para la generación de la autorización, si los datos son validos el sistema crea un registro en la base de datos para que cuando el visitante requiera ingresar el guarda lo registre.			
3.2 ACTORES			
Propietario de oficina			
3.3 PRECONDICIONES			
1. El propietario debe estar dado de alta en el sistema y poseer una contraseña de autorización.			

2. La oficina debe estar dada de alta en el sistema
3. El sistema debe estar operativo

3.4 FLUJO NORMAL

Contando con las precondiciones el flujo normal será el siguiente:

Paso	Actor	Sistema
1	El propietario ingresa al modulo de autorización de visitante	El sistema muestra el formulario de autorización con los campos oficina, id visitante, nombre quien autoriza, fecha de ingreso y contraseña de autorización
2	El propietario escoge oficina, llena los campos correspondientes y pulsa el botón autorizar	El sistema valida que todos los campos estén llenos o escogidos y la contraseña sea correcta
3		Si las validaciones son correctas el sistema crea un registro en la tabla visitantes con los datos llenados y deja null los campos que no son entregados(nombre, apellido y hora de ingreso)
4		El sistema muestra el mensaje “Autorización registrada correctamente para el documento XXXX”

3.5 FLUJO ALTERNATIVO

Si existe otra forma de acceder al caso de uso, describir los pasos.

Paso	Actor	Sistema
------	-------	---------

3.5 FLUJO EXCEPCIONAL

Paso	Actor	Sistema
2A1		Si los datos están incompletos, no corresponden con el tipo o no son escogidos, el sistema muestra el mensaje “Campo incompleto o incorrecto, verifique”
2A2	El propietario debe llenar el formulario correctamente y pulsa el botón autorizar	El sistema valida que todos los campos estén llenos o escogidos y la contraseña sea correcta
2B1		Si la contraseña es incorrecta, el sistema muestra el mensaje “Contraseña incorrecta, verifique”
2B2	El propietario debe escribir la contraseña correctamente y	El sistema valida que todos los campos estén llenos o escogidos y la contraseña sea correcta

	pulsar el botón autorizar	
3.1		Si, el sistema encuentra un error al guardar el registro mostrara el mensaje “Error guardando el registro, intente nuevamente o contacte a soporte”
3.2	El propietario debe intentar nuevamente, si se presenta el error debe contar al soporte	

3.7 POSICIONES

Se crea un registro de autorización en la tabla visitantes con los campos, identificación, oficina_id, autorizo_nombre, fecha_ingreso_autorizado. El registro queda disponible para ser utilizado en el caso CU-02

3.8 FRECUENCIA

Alta

2. IDENTIFICACIÓN DE CASO DE USO

1.1 Id Caso	CU-02	1.2 Nombre	Registrar visitante
-------------	-------	------------	---------------------

3. HISTORICO DE CASO DE USO

2.1 Autor	David Bernal
-----------	--------------

2.2 Fecha Creación	15/11/2025	4. Última Actualización	
--------------------	------------	-------------------------	--

2.4 Actualizado por		2.5 Versión	1.0
---------------------	--	-------------	-----

4. DEFINICIÓN DE UN CASO DE USO**3.1 DESCRIPCIÓN**

Permite al guarda verificar si hay una previa autorización de un visitante y posteriormente registrar el ingreso del visitante. El guarda digita en el campo de identificación del visitante y si este se encuentra en la tabla de visitantes le mostrara los datos de quien autoriza y la oficina a la que se dirige y habilitara los campos para realizar el registro del visitante y acepta para guardar los datos. Si no existe autorización, se mostrara que no hay autorización y su ingreso será denegado.

3.2 ACTORES

Guarda del edificio

3.3 PRECONDICIONES

4. Puede existir o no una autorización de visitante.

5. El sistema debe estar operativo

3.4 FLUJO NORMAL

Contando con las precondiciones el flujo normal será el siguiente:

Paso	Actor	Sistema
1	El guarda ingresa al modulo de registro de visitantes	El sistema muestra el formulario de registro con el campo id visitante

2	El guarda, digita el campo de identificación del visitante y pulsa el botón buscar.	El sistema valida que el campo este lleno
3		Si las validaciones son correctas el sistema busca un registro en la tabla visitantes asociando el número de identificación con el campo identificación
4		Si el sistema encuentra un registro asociado muestra en el formulario la oficina a la que se dirige el visitante, el nombre quien autorizo, la fecha de ingreso autorizado y habilita los campos nombre y apellido del visitante.
5	El guarda debe llenar los campos de nombre y apellido del visitante y oprime el botón Registrar visitante	El sistema validara que los campos estén llenos y actualizara los campos nombre, apellido y hora de ingreso, con la fecha y hora de ingreso del sistema.
6		El sistema muestra el mensaje, “Visitante registrado correctamente”

3.5 FLUJO ALTERNATIVO

Si existe otra forma de acceder al caso de uso, describir los pasos.

Paso	Actor	Sistema
------	-------	---------

3.5 FLUJO EXCEPCIONAL

Paso	Actor	Sistema
2A1		Si los campos están incompletos o no corresponden con el tipo, el sistema muestra el mensaje “Campo incompleto o incorrecto, verifique”
2A2	El guarda debe digitar el id del visitante correctamente y pulsar el botón buscar	El sistema valida que todos los campos estén llenos
4		Si el id del visitante no es encontrado, el sistema muestra el mensaje “No existe autorización con el número de identificación digitado, verifique”, y no habilita campos del formulario
5.1		Si los campos están incompletos, el sistema muestra el mensaje “Campos incompletos, verifique”

5.2	El guarda debe llenar los campos de nombre y apellido del visitante completamente y oprimir el botón Registrar visitante		

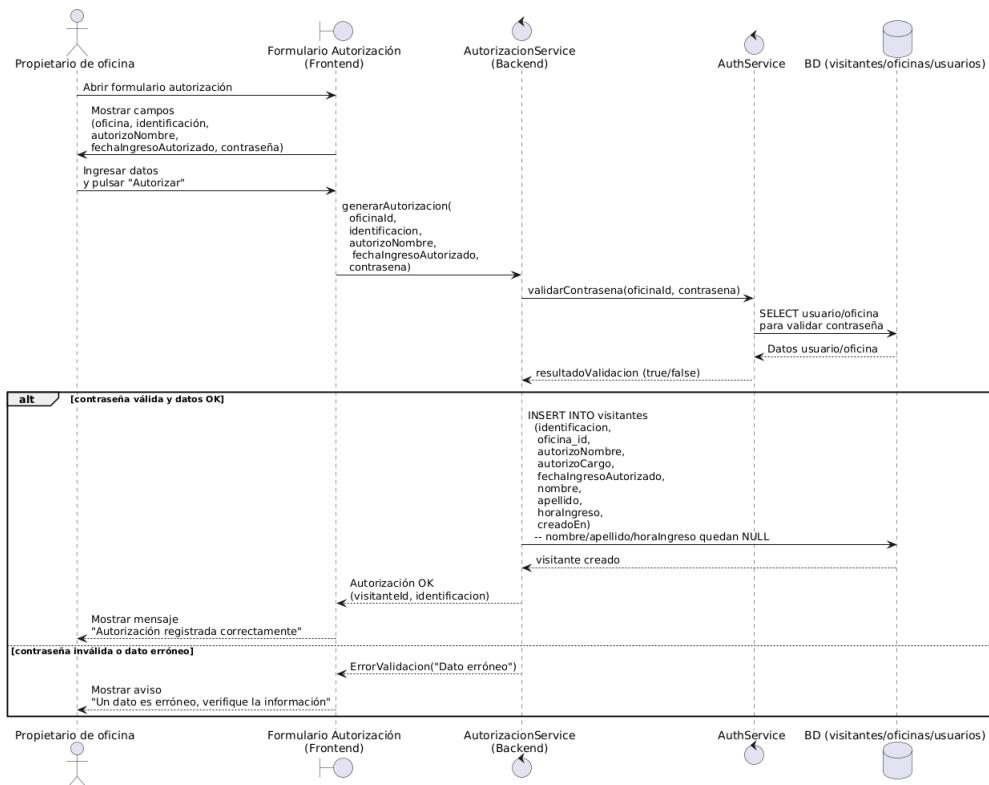
3.7 POS CONDICIONES

Se actualiza el registro de autorización correspondiente en la tabla visitantes los campos, nombre, apellido, hora_ingreso.

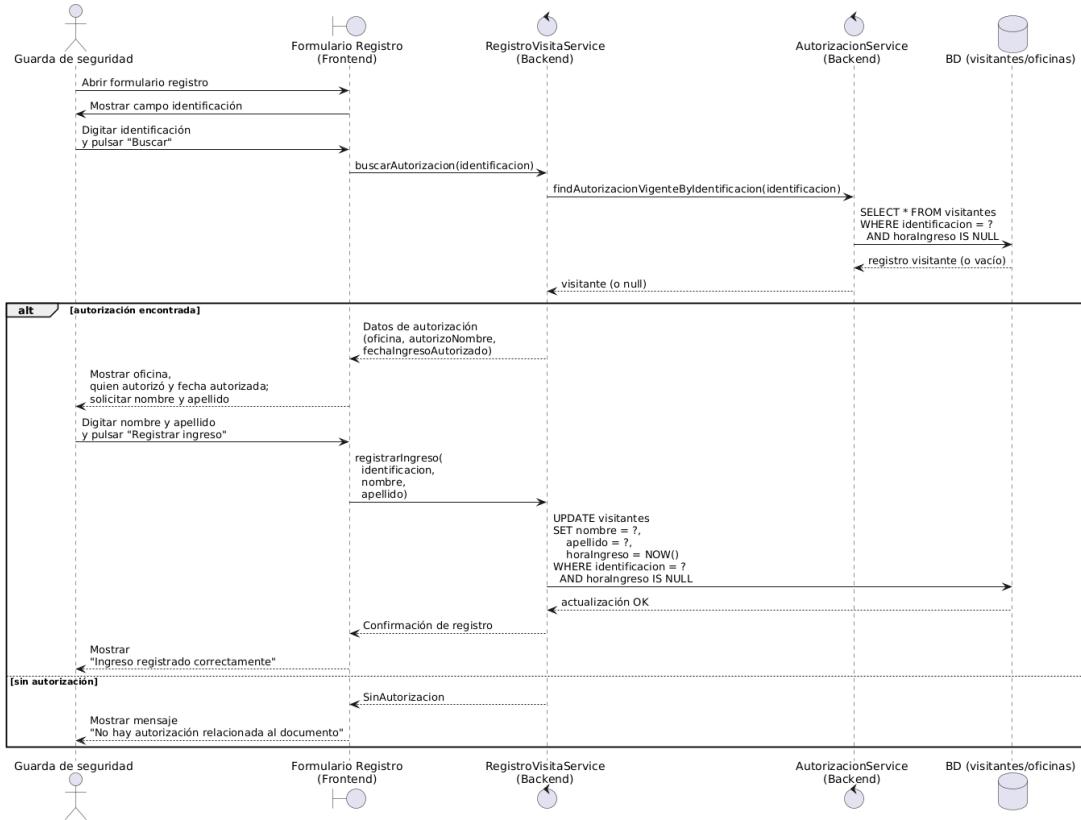
3.8 FRECUENCIA

Alta

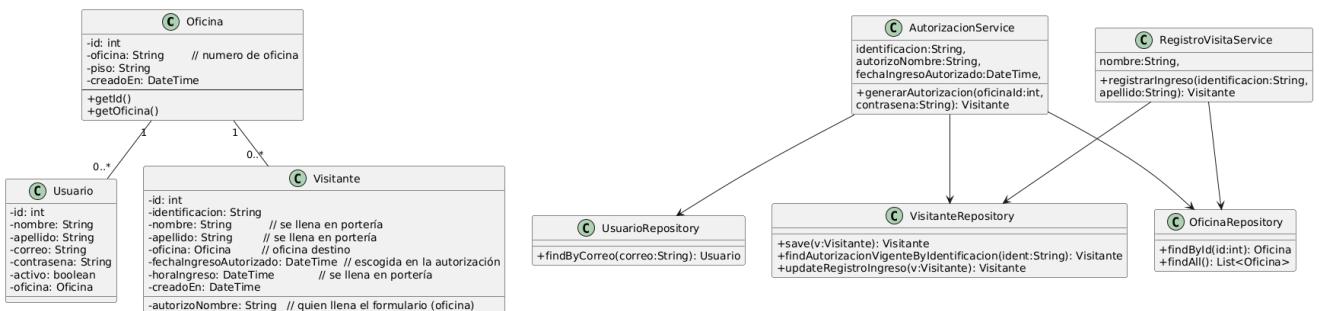
- o Diagrama de secuencia autorización



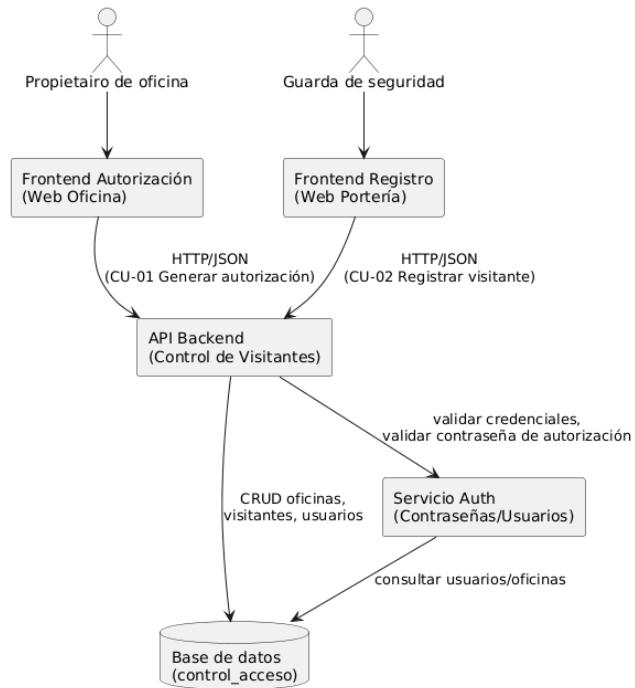
- Diagrama de secuencia registro



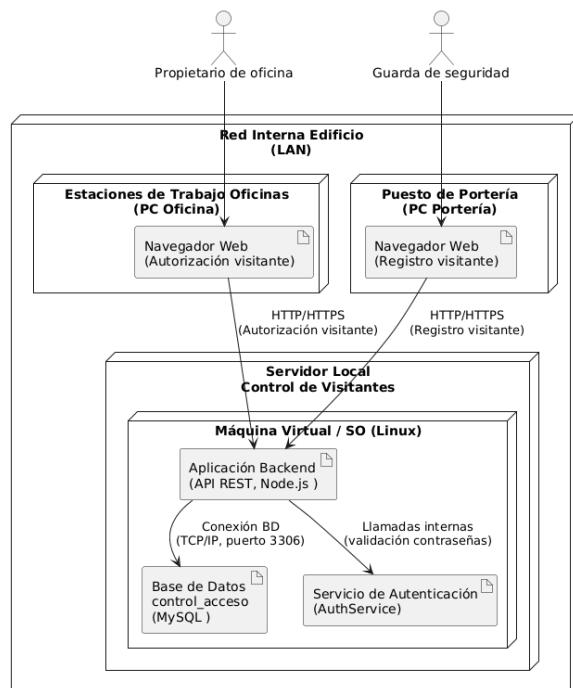
- Diagrama de clases



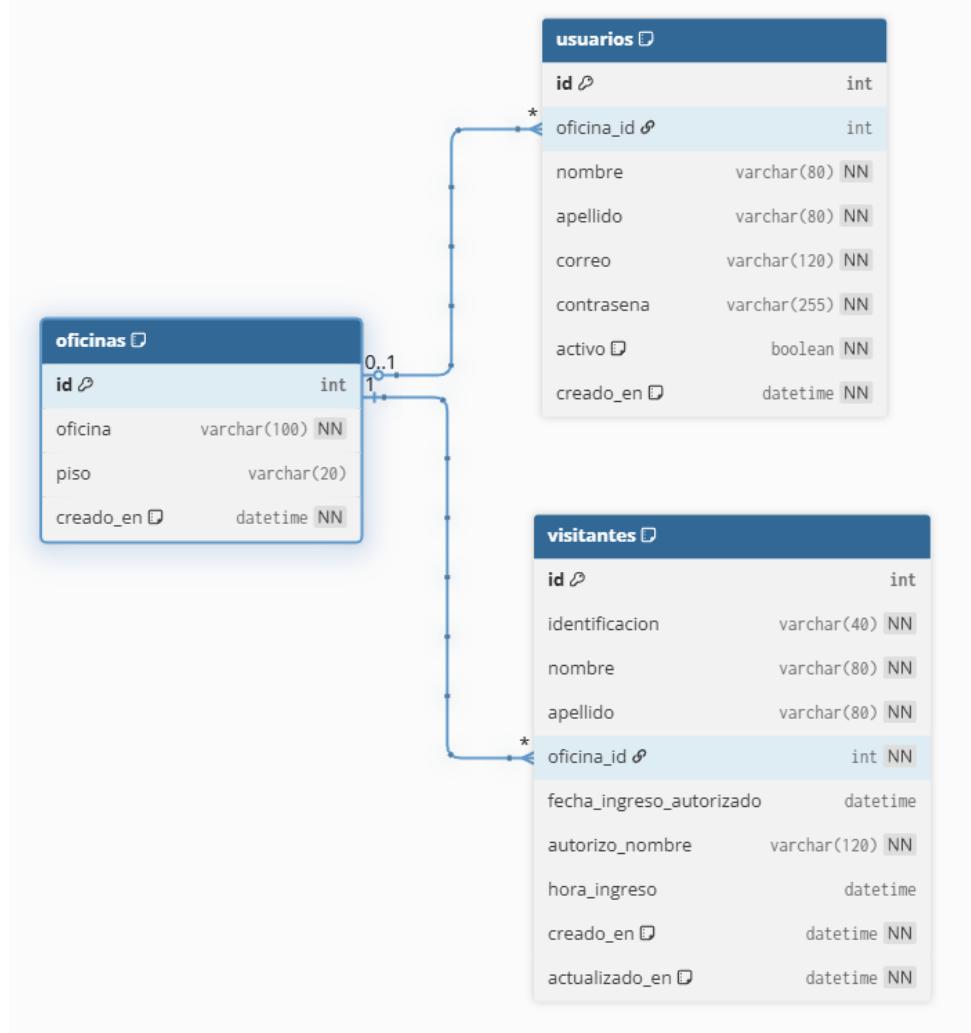
- Diagrama de componentes



- Diagrama de despliegue



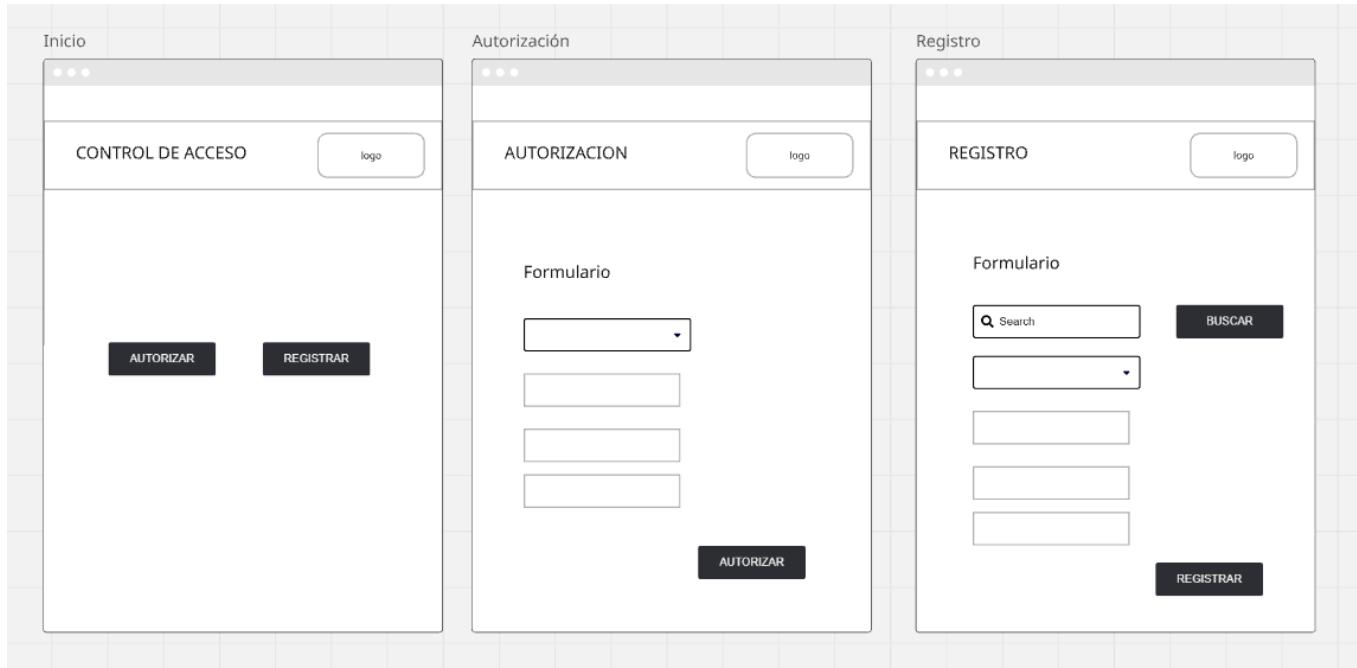
- Modelo de base de datos:



- **Prototipos de interfaz:**

- Baja fidelidad (**Bernal 2025a**)

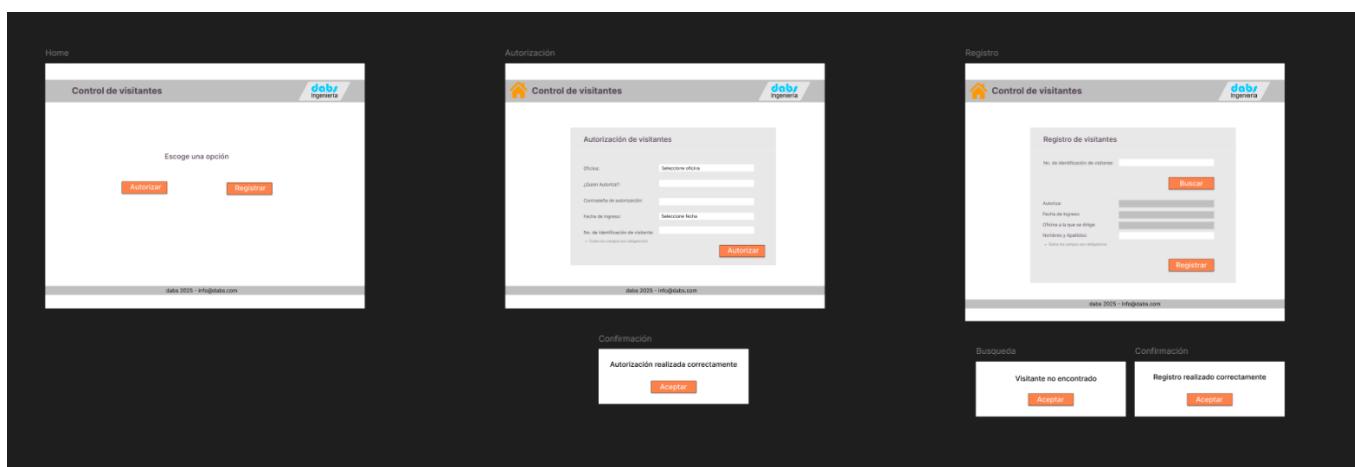
Enlace: https://miro.com/app/board/uXjVI67R-Gg=/?share_link_id=873652450561



- Alta fidelidad: Figma, Bernal, D. (2025a)

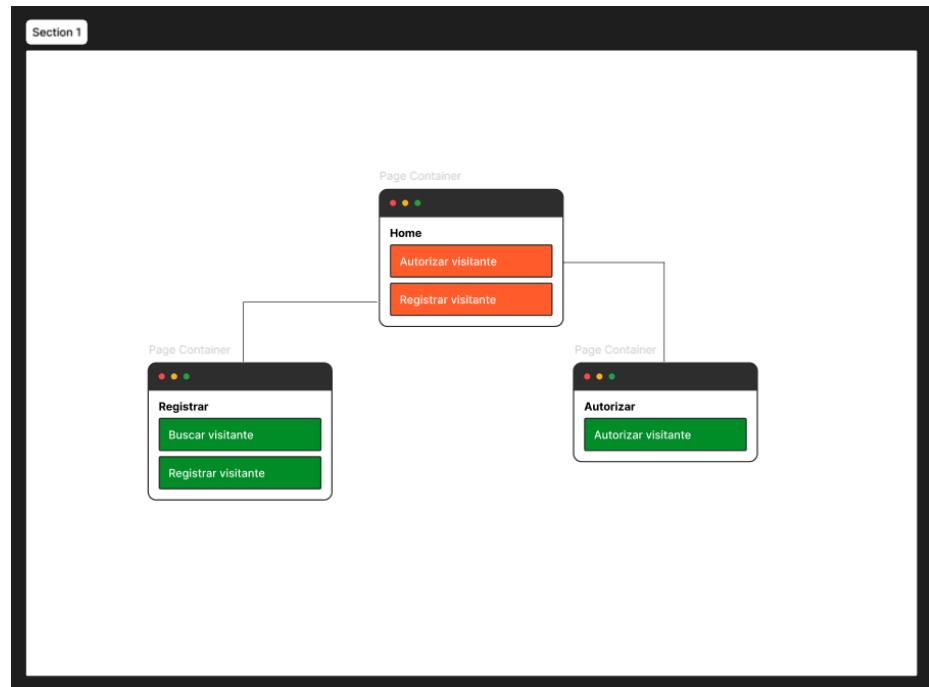
Enlace proyecto: <https://www.figma.com/design/bQAGEGX2vPBiS9sCV2L3OY/Prototipo-control-de-acceso-PS?node-id=2003-9&t=wdbrmNrLvhk9dqKj-1>

Figma Proyecto



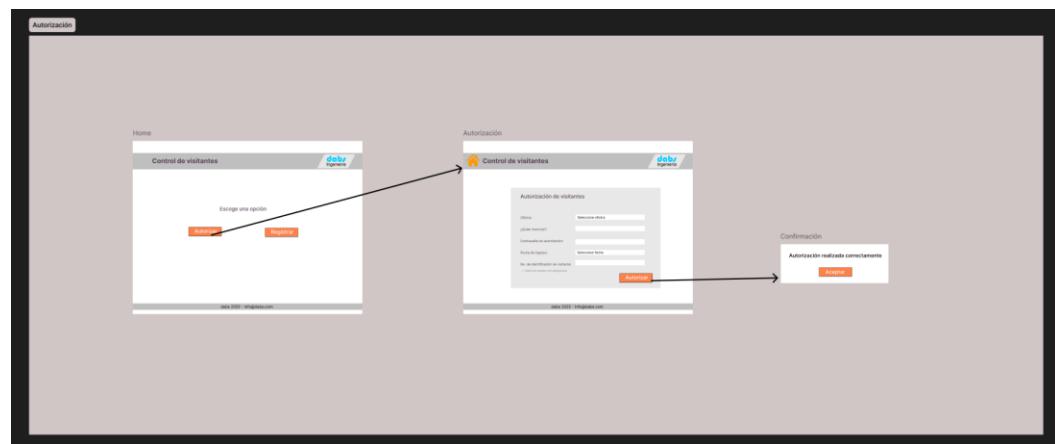
User Map

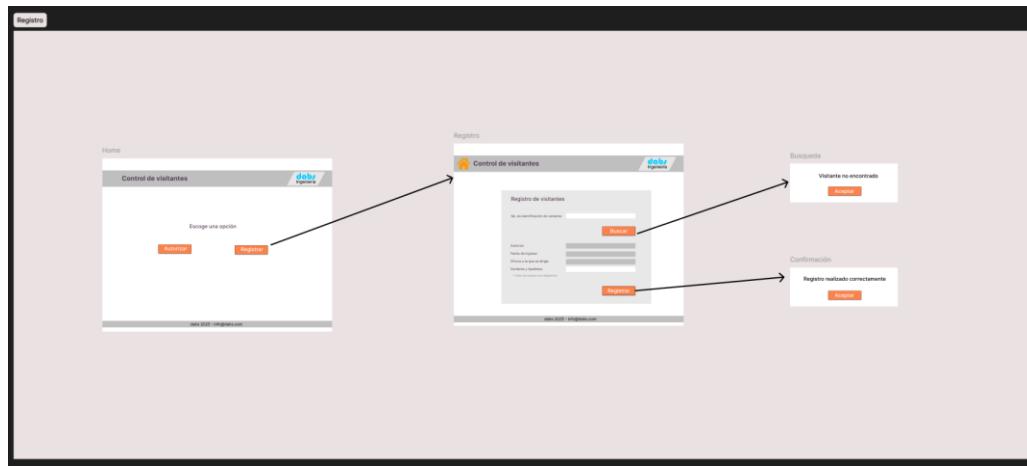
Enlace: <https://www.figma.com/design/bQAGEGX2vPBiS9sCV2L3OY/Prototipo-control-de-acceso-PS?node-id=2011-10&p=f>



UserFlow

Enlace: <https://www.figma.com/design/bQAGEGX2vPBiS9sCV2L3OY/Prototipo-control-de-acceso-PS?node-id=2011-63&t=wdbrmNrLvhk9dqKj-1>





Pruebas de Usabilidad:

- Nombre del proyecto: Prototipo control de acceso PS
- Versión del prototipo: 1.0
- Fecha de la prueba: 15/11/2025
- Equipo responsable: dabs
- Herramientas usadas (Figma, Maze)
- Número de participantes: 5 participantes

Tarea de autorización:

- Objetivos de la prueba
 - 1-Ingresar al módulo de autorización
 - 2-Autorizar un visitante
 - 3-Dejar comentario sobre la aplicación
- Resultados: Tarea 1
 - Tiempo promedio: 3s
 - Tasa de éxito: 80%
 - Errores: 1, no finalizo
- Resultados: Tarea 2
 - Tiempo promedio: 3.3s
 - Tasa de éxito: 80%
 - Errores: 1, no finalizo

- Observaciones

Responses
Click a participant's row to view full session

Muy sencilla	Participant 469628139
no entendi lo que tenia que hacer	Participant 469633253
Muy intuitiva	Participant 469633225
Estuvo facil de realizar	Participant 469633500
todo perfecto	Participant 469633083

Tarea de registro:

- Objetivos de la prueba
 - 1-Ingresar al módulo de registro
 - 2-Buscar un visitante
 - 3-Registrar un visitante
 - 4- Dejar comentario sobre la aplicación
- Resultados: Tarea 1
 - Tiempo promedio: 7.5s
 - Tasa de éxito: 80%
 - Errores: 1, no finalizo
- Resultados: Tarea 2
 - Tiempo promedio: 6.2s
 - Tasa de éxito: 100%
 - Errores: Ninguno
- Resultados: Tarea 3
 - Tiempo promedio: 3.1s
 - Tasa de éxito: 80%
 - Errores: 1, no finalizo

- Observaciones

Responses

La aplicación es muy sencilla

Participant 469633225

estuvo todo bien

Participant 469633582

Enlace Figma ejecutable: <https://www.figma.com/proto/bQAGEGX2vPBiS9sCV2L3OY/Prototipo-control-de-acceso-PS?node-id=2011-132&p=f&t=wdbrmNrLvhk9dqKj-0&scaling=min-zoom&content-scaling=fixed&page-id=2011%3A63>

Enlace Maze - Tarea de registro: <https://t.maze.co/469624401>

Enlace Maze - Tarea de autorización: <https://t.maze.co/469608140>

Patrones y Arquitectura de Software:

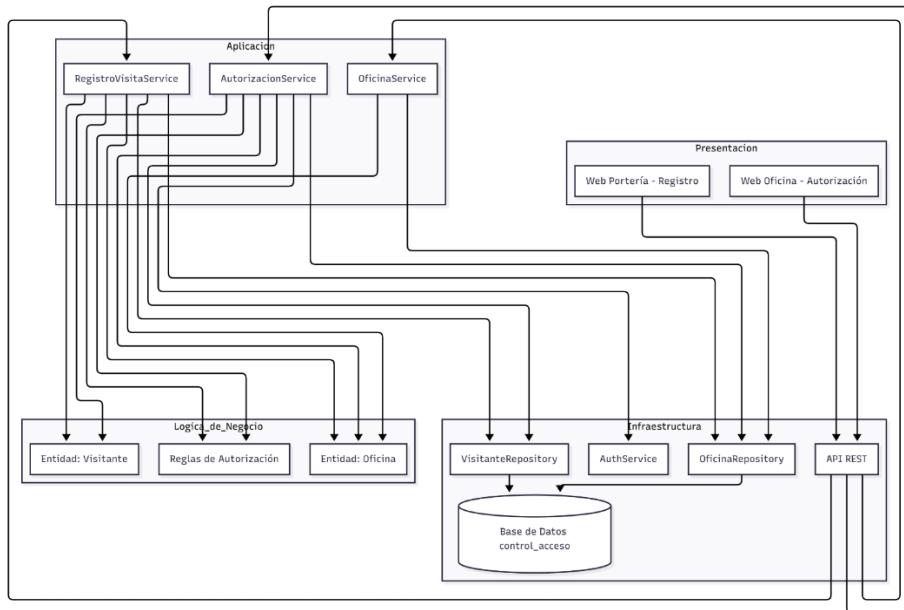
- Patrones arquitectónicos y de diseño aplicados al proyecto.

Seleccione una arquitectura de capas con un patrón arquitectónico cliente-servidor, los clientes en este caso son los usuarios que autorizan o registran visitantes y el servidor será manejado con una API REST, la comunicación será sobre HTTP con JSON y cada operación será un endpoint. Se aplicá un patrón de capa de servicios para encapsular la lógica de los casos de usos y un patrón de repositorio para el acceso a la base de datos mediante consultas SQL, desacoplando los datos de la lógica de negocio

En cuanto a los patrones de diseño aplicados, la tecnología que se va utilizar es node.js con express como backend.

- o Patrón Singleton, estará enfocado a la gestión de la conexión a la base de datos utilizando un pool para reutilizar la instancia mejorando la eficiencia
- o Patrón Factory, garantiza la consistencia que cualquier autorización se cree con todos los campos obligatorios

- Diagrama de arquitectura del desarrollo



- Estructura del proyecto en el repositorio (GitHub, GitLab, etc.).

```

    ▼ CODIGO
      > node_modules
      ▼ SRC
        ▼ public
          > css
          > images
          > js
          > views
          .env
        JS app.js
        JSON package.json
  
```

- Elevator pitch

Enlace: <https://youtu.be/0sgHK7y41hc>

- Repositorio

Enlace: <https://github.com/davidanbernal/ProyectoDeSoftware>

Conclusiones

- La generación de los diagramas permite entender de una forma clara y grafica el flujo de la aplicación, es de suma importancia que todos estos diagramas sean realizados antes de empezar a codificar para que los programadores puedan tener una idea clara de la ejecución, alcance y necesidades de la aplicación.
- La definición de la arquitectura y de los patrones permite determinar cómo van a operar los diferentes modulos garantizado una operación acorde a las necesidades.
- Los prototipos de baja y alta fidelidad permiten tener una idea cercana al producto final permitiendo evaluar su parte gráfica y funcionalidad básica.
- Las pruebas de usabilidad, permiten anteponernos al uso cotidiano por diferentes usuarios y tener una evaluación y retroalimentación de su funcionamiento básico y de toda la elección de operación, colores y forma de uso, logrando tener unas estadísticas para determinar cual podría ser el mejor camino.

Referencias

- Bernal, D. (2025a). Actividad 1,2,3, Análisis y Diseños de sistemas [Manuscrito no publicado]. Universidad Iberoamericana.
- Bernal, D. (2025b). Actividad 3, 4, 5, 6, Gerencia y control de proyectos informáticos [Manuscrito no publicado]. Universidad Iberoamericana.

TERCERA ENTREGA

1. Fase de Desarrollo - Codificación:

- **Arquitectura del software**

La arquitectura implementada es por capas:

1- Capa de presentación

Las vistas son renderizadas a través EJS/HTML

2- Capa cliente

La capa cliente es manejado a través de los scripts registro.js y autorización.js, en ella se hacen todas las validaciones, control y llamados HTTP

3- Capa de aplicación

El servidor recibe en rutas Express en app.js, en este se hace toda la inicialización del controlador, requests, validaciones de datos, mantiene toda la lógica de negocio y devuelve las respuestas en JSON

4- Capa de acceso a datos

En esta se hace los llamados a la base de datos y se realiza las consultas y actualizaciones.

Aunque el controlador y datos, esta en un mismo archivo se realizo de esta manera para mantener un solo archivo debido a que el proyecto es pequeño, para un proyecto mayor estas dos capas deben separarse.

- **Patrones de Diseño**

-Front Controller: Intrínsecamente Express funciona de esta manera, donde todas las peticiones entran por el servidor y se enrutan a un handler

-Data Access: pool.query() encapsula los datos y es la parte encargada de dar el acceso.

-Strategy: Las validaciones realizadas en front se comportan como las reglas aplicadas antes de enviar los datos

- Aplicación de principios SOLID
2. Pruebas de Software y Documentación Estándar
- Pruebas según el Nivel

ID Req	Requisito	ID Caso	Nivel de Prueba
RF-01	Autorizar visitantes validando campos obligatorios	BU-01	Unitaria (Back)
RF-01	Autorizar visitantes validando credenciales	BU-02	Unitaria (Back)
RF-01	Registrar autorización válida	BU-04	Unitaria (Back)
RF-02	Evitar autorizaciones duplicadas por día	BU-03	Unitaria (Back)
RF-02	Evitar duplicados en concurrencia	NF-02	Integración
RF-03	Registrar ingreso solo con autorización previa	INT-03	Integración
RF-03	Flujo completo autorización + registro	SIS-01	Sistema
RF-04	Cargar oficinas dinámicamente desde BD	INT-01	Integración
RF-05	Validar fecha de ingreso (hoy o posterior)	FU-02	Unitaria (Front)
RF-06	Habilitar botones según estado del flujo	FU-03	Unitaria (Front)
RF-07	Mostrar mensajes claros al usuario	UAT-03	Aceptación
RF-08	Garantizar respuesta ante fallas de BD	SIS-02	Sistema
RF-09	Desempeño aceptable del sistema	NF-01	Sistema

- Pruebas según la Técnica

ID Req	Requisito	ID Caso	Técnica
RF-01	Autorizar visitantes validando campos obligatorios	BU-01	Caja blanca
RF-01	Autorizar visitantes validando credenciales	BU-02	Caja blanca
RF-01	Registrar autorización válida	BU-04	Caja blanca
RF-02	Evitar autorizaciones duplicadas por día	BU-03	Caja blanca
RF-02	Evitar duplicados en concurrencia	NF-02	Caja gris
RF-03	Registrar ingreso solo con autorización previa	INT-03	Caja gris
RF-03	Flujo completo autorización + registro	SIS-01	Caja negra
RF-04	Cargar oficinas dinámicamente desde BD	INT-01	Caja gris
RF-05	Validar fecha de ingreso (hoy o posterior)	FU-02	Caja negra
RF-06	Habilitar botones según estado del flujo	FU-03	Caja negra
RF-07	Mostrar mensajes claros al usuario	UAT-03	Caja negra
RF-08	Garantizar respuesta ante fallas de BD	SIS-02	Caja negra
RF-09	Desempeño aceptable del sistema	NF-01	Caja negra

- **Pruebas de Funcionalidad y Pruebas No Funcionales**

ID Req	Requisito	ID Caso	Tipo
RF-01	Autorizar visitantes validando campos obligatorios	BU-01	Funcional
RF-01	Autorizar visitantes validando credenciales	BU-02	Funcional
RF-01	Registrar autorización válida	BU-04	Funcional
RF-02	Evitar autorizaciones duplicadas por día	BU-03	Funcional
RF-02	Evitar duplicados en concurrencia	NF-02	No funcional (confiabilidad)
RF-03	Registrar ingreso solo con autorización previa	INT-03	Funcional
RF-03	Flujo completo autorización + registro	SIS-01	Funcional
RF-04	Cargar oficinas dinámicamente desde BD	INT-01	Funcional
RF-05	Validar fecha de ingreso (hoy o posterior)	FU-02	Funcional
RF-06	Habilitar botones según estado del flujo	FU-03	Funcional
RF-07	Mostrar mensajes claros al usuario	UAT-03	No funcional (usabilidad)
RF-08	Garantizar respuesta ante fallas de BD	SIS-02	No funcional (disponibilidad)
RF-09	Desempeño aceptable del sistema	NF-01	No funcional (rendimiento)

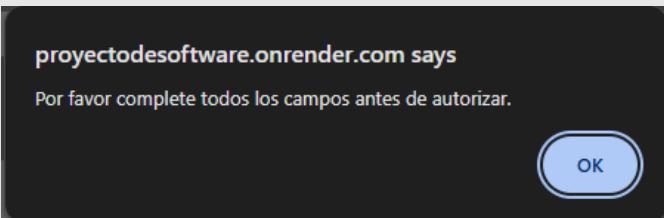
- Documentación Casos de Pruebas

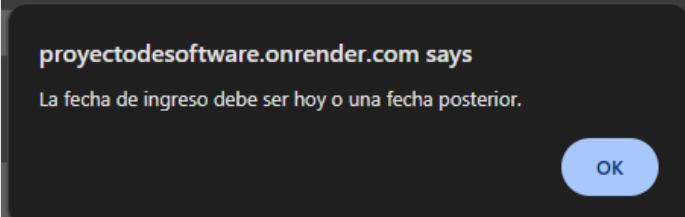
Validación de campos obligatorios en autorización	BU-01	
	¿Prueba de despliegue?	No
Descripción: Verificar que el sistema impida registrar una autorización cuando uno o más campos obligatorios no han sido diligenciados.		
Prerrequisitos 1. El servidor backend está en ejecución. 2. La base de datos está disponible.		
Pasos: Solicitud POST /autorizar con al menos un campo vacío.		
Resultado esperado: - El sistema rechaza la solicitud. - Se retorna un mensaje indicando que todos los campos son obligatorios.		
Resultado obtenido: El servidor respondió con código 400 y no se hizo ninguna inserción en la BD { "success": false, "message": "Todos los campos son obligatorios." }		

Validación de oficina y contraseña incorrecta	BU-02	
	¿Prueba de despliegue?	No
Descripción: Validación de oficina y contraseña incorrecta		
Prerrequisitos - Existe al menos una oficina registrada en la base de datos. - Existe un usuario asociado a una oficina con una contraseña válida.		
Pasos: Solicitud POST /autorizar con oficina válida y contraseña incorrecta		
Resultado esperado: - La autorización es rechazada. - El sistema retorna un mensaje de error informativo.		
Resultado obtenido: El servidor respondió con código 200 y no se hizo ninguna inserción en la BD porque la oficina o la contraseña no eran correctas { "success": false, "message": "Los datos de oficina y contraseña no son correctos. Vuelva a intentarlo." }		

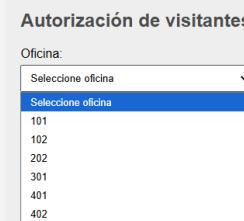
Validación de autorización duplicada	BU-03	
	¿Prueba de despliegue?	No
Descripción: Verificar que el sistema no permita registrar más de una autorización para el mismo visitante en la misma fecha.		
Prerrequisitos - Existe un registro previo en la tabla visitantes con la misma identificación y fecha.		
Pasos: -Solicitud POST /autorizar con identificación y fecha ya registradas.		
Resultado esperado: - El sistema rechaza la autorización. - Se retorna un mensaje indicando que ya existe una autorización para ese día		
Resultado obtenido: El servidor respondió con código 200 y no se hizo ninguna inserción en la BD porque en la fecha seleccionada ya hay un registro con el mismo documento. { "success": false, "message": "Ya hay una autorización registrada para este documento en ese día." }		

Registro exitoso de autorización	BU-04	
	¿Prueba de despliegue?	No
Descripción: Verificar que el sistema registre correctamente una autorización cuando todos los datos son válidos.		
Prerrequisitos -La oficina existe. -La contraseña corresponde a la oficina. -No existe una autorización previa para la misma identificación y fecha.		
Pasos: -Solicitud POST /autorizar con datos válidos		
Resultado esperado: -Se inserta un registro en la tabla visitantes. -El sistema retorna un mensaje de éxito.		
Resultado obtenido: El servidor respondió con código 200 y se hizo inserción en la fecha seleccionada con los datos digitados { "success": true, "message": "La autorización fue realizada correctamente." }		

Validación de campos vacíos en formulario	FU-01	
	¿Prueba de despliegue?	No
Descripción: Verificar que el sistema no permita enviar el formulario de autorización si existen campos obligatorios vacíos.		
Prerrequisitos -El usuario se encuentra en la página de autorización.		
Pasos: Formulario con uno o más campos vacíos.		
Resultado esperado: -Se muestra un mensaje de alerta. -No se envía la solicitud al backend.		
Resultado obtenido: Se realiza intento de enviar el formulario con campos incompletos y muestra el mensaje que deben ser completados antes de continuar		
		

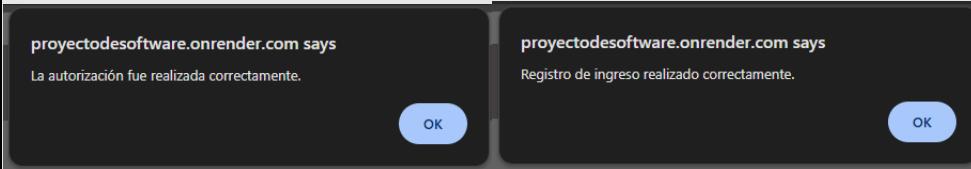
Validación de fecha de ingreso	FU-02	
	¿Prueba de despliegue?	No
Descripción: Validar que el sistema no permita seleccionar una fecha anterior a la fecha actual		
Prerrequisitos -El usuario se encuentra en la página de autorización.		
Pasos: -Ingresar fecha anterior a la fecha actual.		
Resultado esperado: -Se muestra una alerta indicando que la fecha no es válida. - El formulario no se envía.		
Resultado obtenido: Se realiza intento de enviar el formulario con fecha anterior a la actual y muestra el mensaje que debe ser seleccionada una fecha posterior antes de continuar		
		

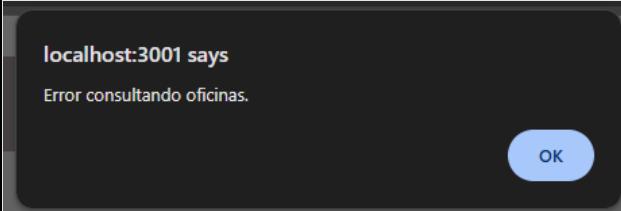
Habilitación del botón Registrar	FU-03	
	¿Prueba de despliegue?	No
Descripción: Verificar que el botón <i>Registrar</i> se habilite únicamente cuando la búsqueda de autorización ha sido exitosa y el campo de nombres está diligenciado.		
Prerrequisitos -El usuario se encuentra en la página de registro. -Existe una autorización válida para el visitante.		
Pasos: Ingresar nombres y apellidos después de una búsqueda exitosa.		
Resultado esperado: -El botón Registrar cambia a estado habilitado.		
Resultado obtenido: Se realiza búsqueda con documento autorizado en la fecha actual y se habilita el botón registrar cuando se ha llenado el campo nombre.		
		

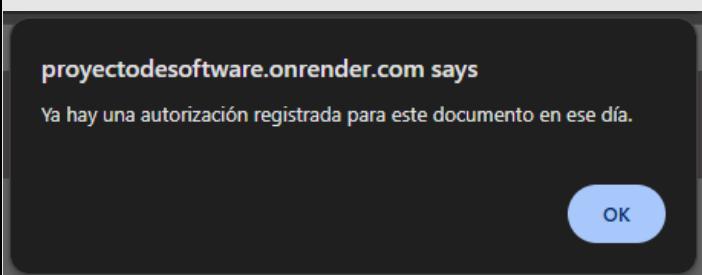
Carga dinámica de oficinas	INT-01	
	¿Prueba de despliegue?	No
Descripción: Validar la integración entre frontend y backend para la carga dinámica de oficinas.		
Prerrequisitos - La tabla oficinas contiene registros. - El backend está disponible.		
Pasos: -Abrir página de autorización.		
Resultado esperado: -El listado de oficinas se muestra correctamente en el <select>.		
Resultado obtenido: Se abre la pagina de autorización de visitantes y al abrir el select de oficinas se muestran correctamente las que están registradas en la BD		
		

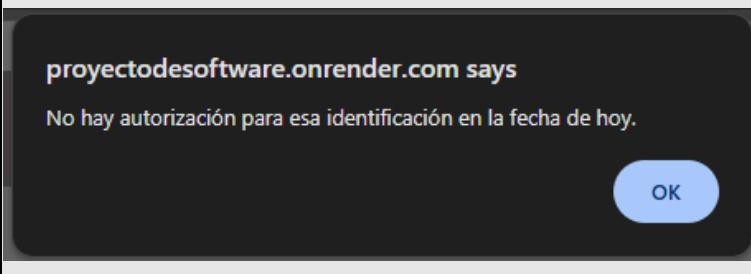
Flujo completo de autorización	INT-02	
	¿Prueba de despliegue?	No
Descripción: Verificar la interacción correcta entre frontend, backend y base de datos durante la autorización.		
Prerrequisitos - Sistema operativo. - Base de datos disponible.		
Pasos: -Llenar formulario de autorización con datos válidos.		
Resultado esperado: - Autorización registrada. - Mensaje de confirmación visible.		
Resultado obtenido: Se realiza proceso completo de autorización de un visitante y el resultado es correcto.		

Registro de ingreso del visitante	INT-03																			
	¿Prueba de despliegue?	No																		
Descripción: Validar la actualización de los datos del visitante luego de una búsqueda exitosa.																				
Prerrequisitos -Existe una autorización válida para el día actual.																				
Pasos: Ingresar nombres y confirmación del registro de una búsqueda exitosa.																				
Resultado esperado: Se actualizan los campos nombre y hora_ingreso.																				
Resultado obtenido: Se realiza proceso completo de registro de un visitante previamente autorizado para este día y los datos son actualizados correctamente en la BD																				
 <table border="1"> <thead> <tr> <th>id</th> <th>identificacion</th> <th>nombre</th> <th>oficina_id</th> <th>oficina</th> <th>fecha_ingreso_autorizado</th> <th>autorizo_nombre</th> <th>hora_ingreso</th> <th>creado_en</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>789456</td> <td>juan zuluaga</td> <td>4</td> <td>401</td> <td>2025-12-12 00:00:00</td> <td>marlen quiroga</td> <td>2025-12-12 23:19:03</td> <td>2025-12-12 23:18:37</td> </tr> </tbody> </table>			id	identificacion	nombre	oficina_id	oficina	fecha_ingreso_autorizado	autorizo_nombre	hora_ingreso	creado_en	1	789456	juan zuluaga	4	401	2025-12-12 00:00:00	marlen quiroga	2025-12-12 23:19:03	2025-12-12 23:18:37
id	identificacion	nombre	oficina_id	oficina	fecha_ingreso_autorizado	autorizo_nombre	hora_ingreso	creado_en												
1	789456	juan zuluaga	4	401	2025-12-12 00:00:00	marlen quiroga	2025-12-12 23:19:03	2025-12-12 23:18:37												

Flujo completo del sistema	SIS-01	
	¿Prueba de despliegue?	No
Descripción: Evaluar el comportamiento del sistema desde la autorización hasta el registro del visitante.		
Prerrequisitos -Sistema desplegado correctamente.		
Pasos: -Ejecutar completamente del flujo por un usuario real.		
Resultado esperado: - Flujo continuo sin errores. - Datos consistentes en la base de datos.		
Resultado obtenido: Se realiza proceso completo de autorización y registro de un visitante y el resultado es correcto.		
		

Manejo de fallos del sistema	SIS-02	
	¿Prueba de despliegue?	No
Descripción: Evaluar la respuesta del sistema ante una falla de la base de datos.		
Prerrequisitos -El sistema está en ejecución.		
Pasos: -Interrumpir la conexión a la base de datos. -Ingresar a autorización de visitantes		
Resultado esperado: - Mensaje de error consultando oficinas. - El sistema no se detiene abruptamente.		
Resultado obtenido: Se abre pagina de autorización de visitantes y al cargar muestra el error consultando oficinas, sin embargo, la página carga correctamente		
		

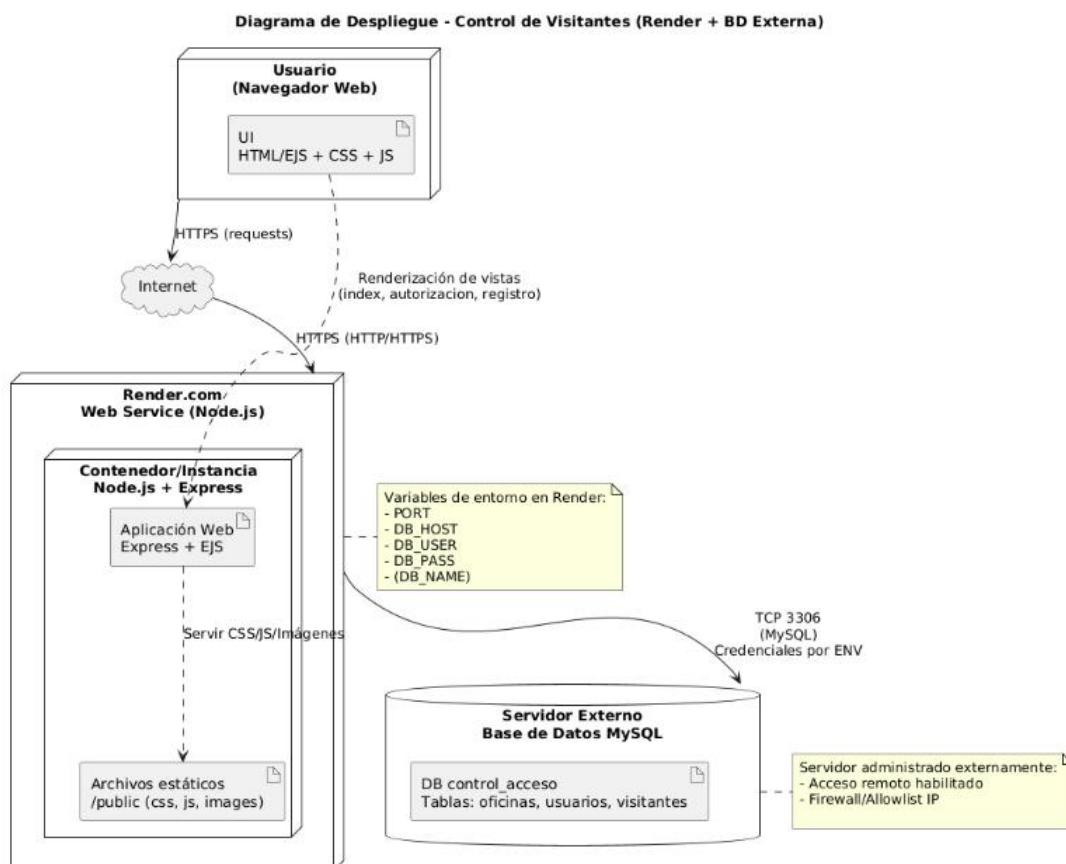
Autorización única por día	UAT-01	
	¿Prueba de despliegue?	No
Descripción: Validar que el sistema cumpla con la restricción de una sola autorización por visitante al día.		
Prerrequisitos -Existe una autorización previa registrada para el visitante en la fecha actual.		
Pasos: -Realizar un Intento de segunda autorización el mismo día.		
Resultado esperado: - El sistema rechaza la autorización duplicada.		
Resultado obtenido: Se realiza proceso de autorización para el misma con una autorización registrada previamente en el sistema, el sistema retorna el mensaje y no almacena el registro en la BD		
		

Registro solo con autorización previa	UAT-02	
	¿Prueba de despliegue?	No
Descripción: Validar que solo se permita el registro de ingreso cuando existe una autorización válida.		
Prerrequisitos -No existe autorización previa		
Pasos: - Ejecutar completa del flujo dos veces de autorización con fecha actual		
Resultado esperado: - El sistema bloquea el registro.		
Resultado obtenido: En la pagina de registro, al buscar una identificación que no tiene autorización para el día actual, el mensaje muestra que no hay autorización		
		

Usabilidad del sistema	UAT-03	
	¿Prueba de despliegue?	No
Descripción: Evaluar la claridad de la interfaz y los mensajes del sistema.		
Prerrequisitos -Usuario final con acceso al sistema.		
Pasos: -Realizar interacción normal con el sistema.		
Resultado esperado: - El usuario comprende el flujo y los mensajes mostrados.		
Resultado obtenido: Después de hacer todos los procesos se evidencia que los mensajes son claros y explicativos ante errores, confirmaciones o advertencias		

3. Proceso de Despliegue: Hacer accesible la aplicación al público.

- **Diagrama de Despliegue**



- **Proceso de despliegue.**

- 1- Creamos la base de datos SQL, para esto vamos a utilizar un servidor externo de un servicio pago, para esto digitamos el nombre, usuario y la contraseña y creamos la base de datos

+ Crear nueva base de datos MySQL y usuario de base de datos

Nombre de la base de datos
MySQL u298424173_ Nombre de la base de datos

Nombre de usuario MySQL u298424173_ Nombre de usuario

Contraseña * o

✓ Crear

- 2- Accedemos al panel de administración de la base de datos e importamos el modelo realizado en local

Importing into the database "u298424173_control_acceso"

File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in **[format].[compression]**. Example: **.sql.zip**

Browse your computer: (Max: 256MiB)

Choose File No file chosen

You may also drag and drop a file on any page.

Character set of the file:

utf-8

- 3- Confirmamos que la estructura y datos precargados hayan sido importados.

Table	Action	Rows	Type	Collation	Size	Overhead
oficinas	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_unicode_ci	32.0 Kib	-
usuarios	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_unicode_ci	48.0 Kib	-
visitantes	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_unicode_ci	64.0 Kib	-
3 tables	Sum	18	InnoDB	utf8mb4_unicode_ci	144.0 Kib	0 B

- 4- Para la aplicación, vamos a utilizar el servicio de render.com, en la página, realizamos el ingreso, escogemos la opción Web service y creamos un proyecto, con esto vamos a tener un formulario donde llamamos el repositorio donde se subió el código, escogemos el lenguaje que para mi caso es node.js, escogemos el Branch del repositorio y el directorio root donde está el código.

5-

Configure and deploy your new Web Service

Choose service > Configure > Deploy Need help? Docs

Source Code

davidanbernal / ProyectoDeSoftware • 3m ago Edit

Name: ProyectoDeSoftware

Language: Node

Branch: main

Region: Oregon (US West)

Root Directory: src

- 6- Escogemos los recursos que necesitamos, para este caso escogemos el servicio gratuito

Instance Type

For hobby projects		Upgrades to enable more features	
Free	512 MB (RAM) \$0 / month	0.1 CPU	Free instances spin down after periods of inactivity. They do not support SSH access, scaling, one-off jobs, or persistent disks. Select any paid instance type to enable these features.
For professional use			
For more power and to get the most out of Render, we recommend using one of our paid instance types. All paid instances support:		Starter \$7 / month	512 MB (RAM) 0.5 CPU
<ul style="list-style-type: none"> ▪ Zero Downtime ▪ SSH Access ▪ Scaling ▪ One-off jobs ▪ Support for persistent disks 		Standard \$25 / month	2 GB (RAM) 1 CPU
		Pro \$85 / month	4 GB (RAM) 2 CPU
		Pro Plus \$175 / month	8 GB (RAM) 4 CPU
		Pro Max \$225 / month	16 GB (RAM) 4 CPU
		Pro Ultra \$450 / month	32 GB (RAM) 8 CPU

Need a custom instance type? We support up to 512 GB RAM and 64 CPUs.

- 7- Se declaran las variables de configuración que fueron declaradas en el archivo. env utilizadas en local y damos click en el botón de despliegue.

Environment Variables

Set environment-specific config and secrets (such as API keys), then read those values from your code. [Learn more](#).

PORT	✖
DB_HOST	✖
DB_USER	✖
DB_PASS	✖

+ Add Environment Variable [Add from .env](#)

8- El Sistema instala las dependencias y ejecuta la aplicación

```

December 12, 2025 at 12:57 PM Live
d252533 Create app.js app a src

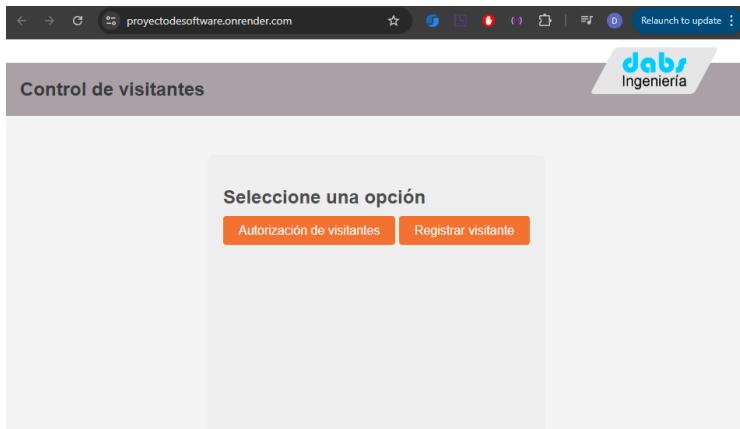
All logs ▾ Search Dec 12, 12:56 PM - 1:03 PM GMT-5 ↗ ...
```

Dec 12 12:59:25 PM npm audit fix --force
 Dec 12 12:59:25 PM Run 'npm audit' for details.
 Dec 12 12:59:28 PM >>> Uploading build...
 Dec 12 12:59:37 PM >>> Uploaded in 4.5s. Compression took 4.0s
 Dec 12 12:59:37 PM >>> Build successful! 🎉
 Dec 12 01:02:07 PM >>> Setting WEB_CONCURRENCY=1 by default, based on available CPUs in the instance
 Dec 12 01:02:07 PM >>> Deploying...
 Dec 12 01:02:46 PM >>> Running 'node app.js'
 Dec 12 01:02:48 PM Servidor corriendo en <http://localhost:3001>
 Dec 12 01:02:48 PM Conectado a la base de datos MySQL
 Dec 12 01:02:51 PM >>> Your service is live! 🎉
 Dec 12 01:02:51 PM >>>
 Dec 12 01:02:51 PM >>> /////////////////////////////////
 Dec 12 01:02:51 PM >>>
 Dec 12 01:02:51 PM >>> Available at your primary URL <https://proyectodesoftware.onrender.com>

9- En la Dirección asignada por la plataforma se desplegará la aplicación para ejecución, al ingresar por primera, el sistema tomará un tiempo en mostrar la aplicación mientras carga todos los componentes.

10- Se realiza ejecución de todos los elementos y se realizan las acciones requeridas.

Página principal



Página de autorización

The screenshot shows a web browser window with the URL "proyectodesoftware.onrender.com/autorizacion". The page title is "Control de visitantes" and features the "dabs Ingeniería" logo. The main content area is titled "Autorización de visitantes" and contains the following fields:

- Oficina: A dropdown menu labeled "Seleccione oficina".
- Quién Autoriza: An input field.
- Contraseña de autorización: An input field.
- Fecha de ingreso: A date input field with the placeholder "mm/dd/yyyy".
- No. de identificación de visitante: An input field.

A small note at the bottom left says "* Todos los campos son obligatorios". At the bottom right is a red "Autorizar" button.

Página de registro

The screenshot shows a web browser window with the URL "proyectodesoftware.onrender.com/registro". The page title is "Control de visitantes" and features the "dabs Ingeniería" logo. The main content area is titled "Registro de visitantes" and contains the following fields:

- No. de identificación de visitante: An input field with a "Buscar" button to its right.
- Autoriza: An input field.
- Fecha de ingreso: An input field.
- Oficina a la que se dirige: An input field.
- Nombres y Apellidos: An input field.

A small note at the bottom left says "* Todos los campos son obligatorios". At the bottom right is a grey "Registrar" button.

-Enlace del Desarrollo Desplegado

<https://proyectodesoftware.onrender.com/>

Para realizar autorización de un visitante la contraseña para cualquier oficina es **12345678**

- Enlace del Video Explicativo Elevetor Pitch

https://youtu.be/RE_cGeUmmLA

- Enlace del Repositorio

<https://github.com/davidanbernal/ProyectoDeSoftware>

- Enlace de base de datos

https://auth-db1847.hstgr.io/index.php?db=u298424173_control_acceso

Datos de acceso de BD

Usuario: u298424173_admin

Contraseña: 3w~R\$&@x@G

Referencias

- Bernal, D. (2025a). Actividad 1,2,3, Análisis y Diseños de sistemas [Manuscrito no publicado]. Universidad Iberoamericana.
- Bernal, D. (2025b). Actividad 3, 4, 5, 6, Gerencia y control de proyectos informáticos [Manuscrito no publicado]. Universidad Iberoamericana.