

ANEXO EXAMEN DE CERTIFICACIÓN

Plan de Estudio	Desarrollo Aplicaciones FullStack Javascript
Caso	Sistema de Gestión de Reservas para Hotel

Prueba: Sistema de Gestión de Reservas para Hotel

¡Felicidades por llegar hasta aquí! Este proyecto es el pináculo de todo lo que has aprendido hasta ahora en el curso. Aquí, pondrás a prueba tus habilidades de desarrollo, para construir un sistema de gestión de reservas que sea funcional, atractivo y adaptable a distintos dispositivos.

Objetivo:

El objetivo de esta prueba es desarrollar una aplicación web completa y dinámica que funcione como un sistema centralizado de gestión de reservas de hoteles. Este sistema permitirá a los usuarios (administrador y huésped) gestionar eficientemente todos los aspectos relacionados con las reservas.

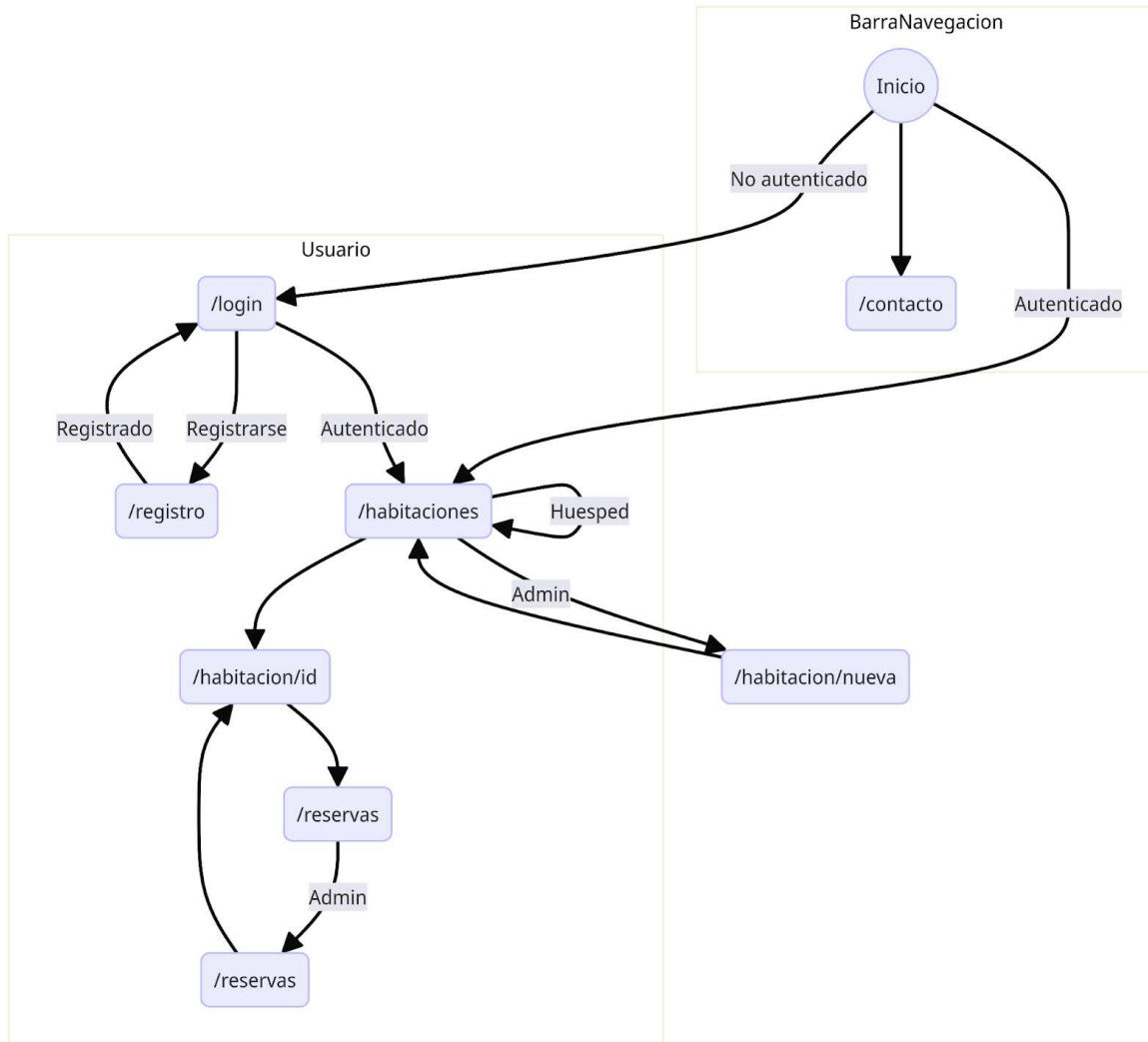
No te preocupes no estarás solo, se ha designado un equipo que te hará entrega de la información necesaria para proceder con el desarrollo del proyecto: información detallada del proyecto, diagrama de flujo, modelo físico y wireframes al final del documento.

Requerimientos funcionales del sistema:

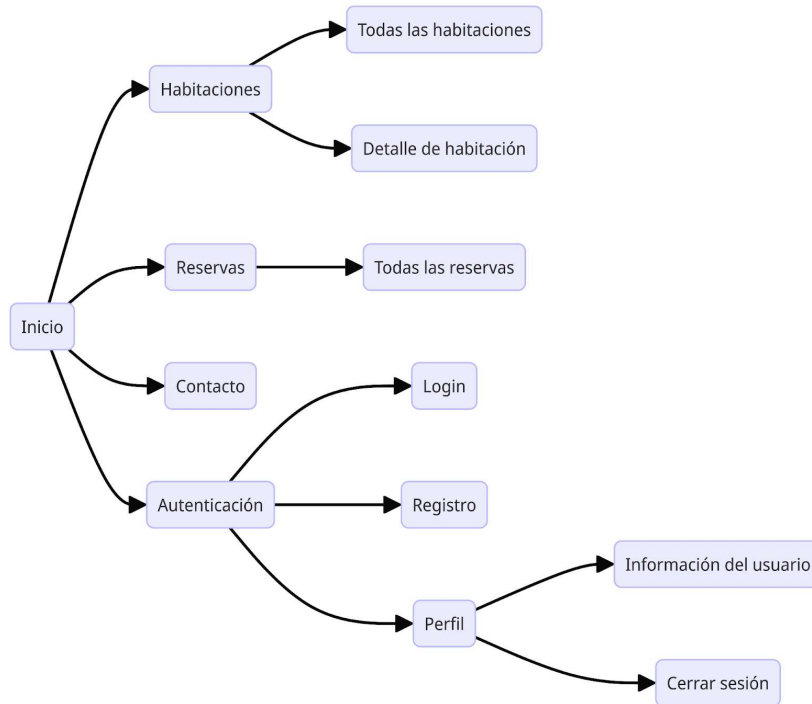
- **Frontend Responsivo:**
 - Utilizar HTML, CSS (Bootstrap) y JavaScript para crear un frontend que se adapte a varios dispositivos.
 - Implementar un diseño intuitivo que facilite la navegación y las interacciones de los usuarios.
- **Modelo de Datos:**
 - **Usuarios:**
 - Detalles como nombre de usuario, correo electrónico, contraseña y tipo de usuario (administrador, huésped). Todo usuario que se registre será un huésped hasta que se le cambie a administrador.
 - **Habitaciones:**

- Información como número de habitación, tipo (simple, doble, suite), descripción, precio y disponibilidad.
- **Reservas:**
 - Información sobre la solicitud de reserva, fechas de entrada, fecha de salida, habitaciones seleccionadas y huésped asociado.
- **Tipos de habitaciones:**
 - Tipos disponibles: simple, doble y suite.
- **Contacto:**
 - Campos para nombre, correo electrónico y mensaje.
- **Funcionalidades:**
 - La ruta raíz (/) debe llevar a la página de **inicio**.
 - Debe existir una vista de **contacto** con su respectivo formulario.
 - Si el usuario no se encuentra autenticado, no podrá ingresar al CRUD de **/habitaciones** y **/reservas**. Si trata de acceder a otras vistas diferentes a **inicio** y **contacto**, será redireccionado a la vista de **inicia sesión**. Al autenticarse, será enviado a la vista **/habitaciones**.
 - Los enlaces de **inicia sesión** y **registrarse** se mostrarán en la barra de navegación si el usuario no se ha autenticado. Al pulsarlos, redireccionará a sus vistas correspondientes.
 - Si el usuario está autenticado, en la barra de navegación se mostrará su **username** y un enlace para **cerrar sesión**, el cual lo redireccionará a la vista de **inicia sesión** al hacer clic en él.
 - La vista de **/habitaciones** debe mostrar 10 de las habitaciones **disponibles** mediante paginación y contar con un filtro donde el usuario podrá filtrar por tipo y precio.
 - Los usuarios de tipo **administrador** podrán generar las habitaciones desde la vista **/habitacion/nueva**, pero el usuario de tipo **huésped** solo tiene permisos para visualizar el listado de habitaciones disponibles.
 - La vista **/habitacion/id** debe mostrar toda la información de la habitación y permitir hacer **reservas** mediante un formulario donde se insertarán la fecha de ingreso y la fecha de salida.
 - Debe existir una vista de **/reservas** en la cual el usuario **administrador** tendrá el listado de las habitaciones reservadas y podrá gestionar el estado de la habitación disponible a **ocupada**.
- **Base de datos:**
 - Crea un archivo llamado **"reservas_hotel.sql"** que contenga el proceso de creación de la base de datos, las tablas con sus restricciones y relaciones.
- **Datos semilla:**
 - Se solicita la creación de datos semilla para las tablas de la base de datos del proyecto. Estos datos pueden estar incluidos dentro del archivo **"reservas_hotel.sql"** o en un archivo separado en formato CSV llamado **"reservas_hotel.csv"**.

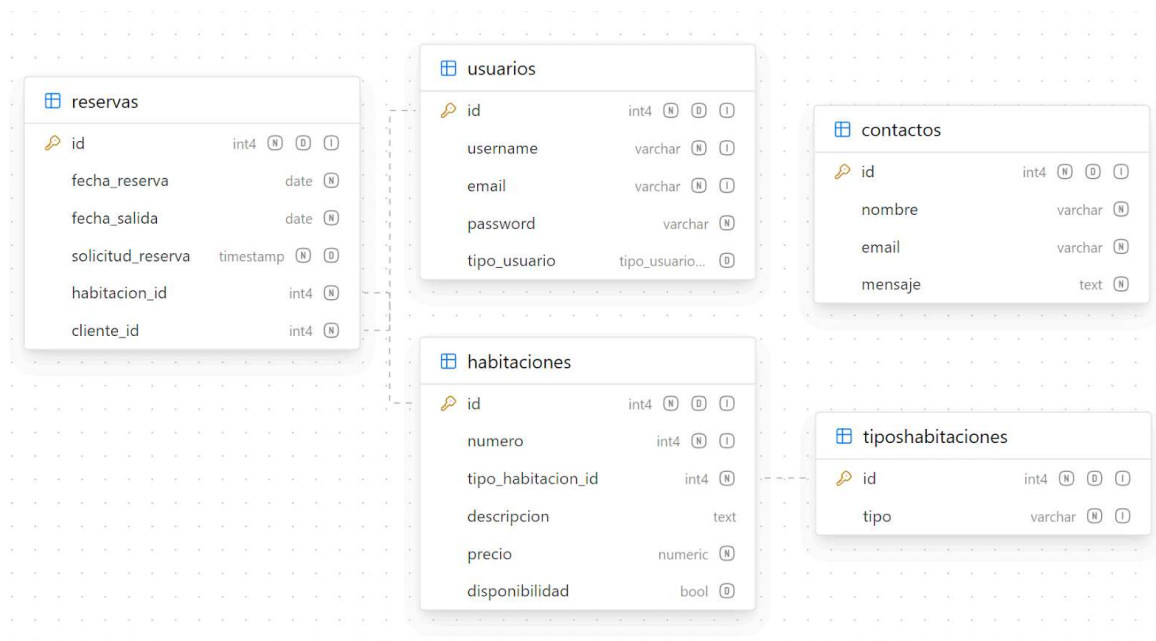
Diagrama de Flujo:



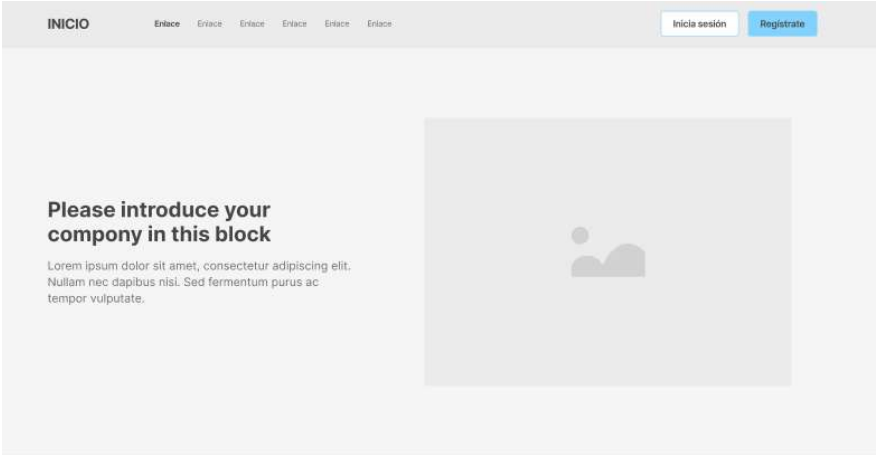
Mapa de sitio:



Modelo de datos:

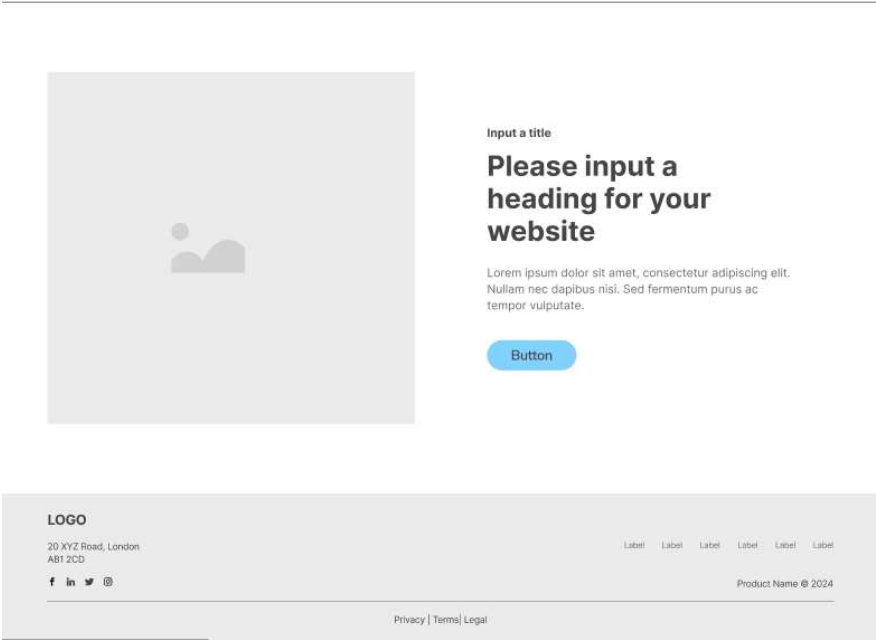


Vistas:



Heading for features page

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam nec dapibus nisi. Sed fermentum purus ac tempor vulputate. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam nec dapibus nisi. Sed fermentum purus ac tempor vulputate.



(/) Inicio/home

Heading for features page

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam nec dapibus nisi. Sed fermentum purus ac tempor vulputate. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam nec dapibus nisi. Sed fermentum purus ac tempor vulputate.

how can we help you?

429 Masonic Drive

contact@myoffice.com

443-663-8698

Nombre

Email

Mensaje

Enviar

LOGO

20 XYZ Road, London
AB1 2CD

f

in

tw

@

Label Label Label Label Label Label

Product Name © 2024

Privacy | Terms | Legal

/contacto

Inicia Sesion

Email

Contraseña

Inicia Sesión

Registro

User

Email

Contraseña

Ingresar

/inicia_sesiony /registerate

INICIO

EnlaceEnlaceEnlaceEnlaceEnlaceEnlace

Pepito Pérez

Tipo de habitación

Fecha

Sub. 15 Jul 2025

Número de personas

Buscar

Tipo de habitación: Doble

\$ 43.000

Numero de habitación: 103

Descripción: Dapibus donec excepteur animi eros vivamus. Numquam tempor accumsan eveniet.

Tipo de habitación: Suite

\$ 80.000

Numero de habitación: 401

Descripción: Dapibus donec excepteur animi eros vivamus. Numquam tempor accumsan eveniet.

Tipo de habitación: Simple

\$ 35.000

Numero de habitación: 208

Descripción: Dapibus donec excepteur animi eros vivamus. Numquam tempor accumsan eveniet.

Tipo de habitación: Suite

\$ 80.000

Numero de habitación: 502

Descripción: Dapibus donec excepteur animi eros vivamus. Numquam tempor accumsan eveniet.

←

010203...20

→

20 XYZ Road, London
AB1 2CD

f

in

LOGO

LabelLabelLabelLabelLabelLabel

Product Name © 2024

Privacy

Terms

Legal

/habitaciones

Crear nueva habitación

Número de habitación

Tipo de habitación ▴

Simple

Doble

Suite

Descripción

Precio

Disponible ☐

Guardar

Número de habitación: 103


Tipo de habitación: Doble


Descripción: Dapibus donec excepteur
animi eros vivamus. Numquam tempor
accumsan eveniet, congue dignissimos,
aut do? Convallis unde cubilia
penatibus? Ut ipsa.

Precio: \$ 43.000

Reservar

Reservar

 Fecha de ingreso
Sab, 14 Jul 2024

 Fecha de salida
Sab, 14 Jul 2024

Julio 2024

M	T	W	T	F	S	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Mensaje

Reservar

hatitacion/id/reserva/

Número de habitación	Huesped	Fecha de ingreso	Fecha de salida	Diponibilidad
101	Juan Ferreira	20/07/2024	27/07/2024	No
402	Romina Gutierrez	25/04/2024	26/04/2024	No
305	Pepito Pérez	17/06/2024	22/06/2024	No
200	Ana Carmiña	02/07/2024	05/07/2024	No
107	Juan Coba	25/05/2024	26/05/2024	No
207	Luisa Chaparro	30/05/2024	03/06/2024	No
302	Esteban Correa	20/08/2024	21/08/2024	No
202	Daniela Castro	08/07/2024	10/08/2024	No
501				Si

Requerimientos no funcionales:

- **Nombre del Proyecto:** reservas_hotel
- **Tecnologías Frontend:**
 - HTML
 - CSS
 - Bootstrap
 - JavaScript
 - jQuery
 - Handlebars.js
- **Tecnologías Backend:**
 - Node.js
 - Express
 - Librerías de Node.js
- **Base de Datos:**
 - PostgreSQL
 - ORM:
 - Se permite el uso de un ORM para la conexión y lógica de la base de datos.
- **Nombre de la Base de Datos:**
 - reservas_hotel
- **Organización del Código:**
 - Mantener separada la lógica de conexión a la base de datos en un archivo.
 - Mantener separadas las consultas a la base de datos.
 - Mantener separada la lógica del servidor.
- **Archivos Requeridos:**
 - package.json
 - package-lock.json
 - reservas_hotel.sql
 - README.md
- **Desarrollo Individual:**
 - El desarrollo de la prueba debe ser individual.
- **Restricciones:**
 - Se prohíbe subir el código a Github y/o compartirlo en plataformas o con otros estudiantes.

Entregables:

1. Archivo Comprimido:

- a. El código fuente del proyecto debe estar comprimido en un archivo con extensión ".zip".
- b. El archivo debe llevar el nombre del proyecto "reservas_hotel.zip", separado por guiones bajos, sin espacios ni mayúsculas.
- c. No se debe incluir la carpeta ".git" ni la carpeta "node_modules" en el archivo comprimido.

2. Archivo SQL:

- a. Se debe incluir un archivo con extensión ".sql" que contenga los comandos para la creación de la base de datos, tablas e inserción de datos.
- b. El archivo debe llamarse "reservas_hotel.sql" y debe ir dentro de la carpeta del proyecto.

3. README.md:

- a. Archivo "README.md" con las siguientes secciones:
 - i. Nombre del Proyecto: Sistema de Gestión de Reservas para Hotel.
 - ii. Descripción del Proyecto: Utiliza el párrafo de objetivos.
 - iii. Capturas de Pantalla del Proyecto: Incluir capturas de pantalla de las siguientes vistas:
 1. Home
 2. Registro
 3. Habitaciones
 4. Habitación/id
 5. Reserva/nueva
 - iv. Prerrequisitos o Dependencias a Instalar.
 - v. Instalación del Proyecto.
 - vi. Comando para Ejecutar el Proyecto.
 - vii. Comando para Cargar la Base de Datos o Migrar los Modelos.
 - viii. Comando para Cargar los Datos Semilla a la Base de Datos.
 - ix. Credenciales de Acceso según Tipo de Usuario luego de Cargar los Datos Semilla:
 1. Para Usuario Tipo Administrador:
 - a. Email: administrador@mail.com
 - b. Contraseña: Abc123#
 2. Para Usuario Tipo Huésped:
 - a. Email: huesped@mail.com
 - b. Contraseña: Abc123#

Puedes usar la siguiente plantilla como referencia para generar tu archivo "README.md":
brayandiazc/simple-readme-template-es: Plantilla README minimalista para proyectos pequeños en español (github.com)

Recomendaciones:

1. Requerimientos y Entregables:

- Lee cuidadosamente cada uno de los requerimientos y entregables antes de comenzar.

2. Creación de la Carpeta:

- Crea una carpeta con el nombre "reservas_hotel" para tu proyecto.

3. Inicialización del Proyecto con Node.js:

- Inicializa tu proyecto con Node.js usando el siguiente comando

```
npm init -y #genera un package.json con información por defecto
```

4. Instalación de Librerías:

- Instala las librerías necesarias con el siguiente comando

```
npm install express nodemon libreria3 libreria4 ...
```

5. Archivos a Generar:

- Genera un archivo server.js para manejar la lógica del servidor.
- Genera un archivo db.js para la configuración de la base de datos.

6. Configuración de la Base de Datos:

- Configura tu conexión a la base de datos PostgreSQL.
- Utiliza llaves primarias en todas las tablas:

```
id SERIAL PRIMARY KEY
```

- Usa las llaves foráneas siempre que sean necesarias.

```
FOREIGN KEY (tabala_uno_id) REFERENCES TablaUno(id)
```

7. Uso de GIT:

- Usa GIT mientras estás desarrollando:

```
git init #para iniciar un proyecto con GIT  
git add . #agrega cambios al proyecto  
git commit -m "mensaje" #agrega los cambios a GIT con un commit  
git checkout id_commit #regresa a una versión anterior del proyecto
```

8. Archivo .gitignore:

- Genera un archivo **.gitignore** con la siguiente configuración:
 - toptal.com/developers/gitignore/api/windows.linux.macos.node

9. Uso de Handlebars.js y Bootstrap:

- Usa Handlebars.js para trabajar con plantillas y componentes en las vistas.
- Utiliza los componentes de Bootstrap en conjunto con CSS para dar estilos a tus vistas. Puedes encontrar la documentación de Bootstrap en [Bootstrap · The most popular HTML, CSS, and JS library in the world.](https://getbootstrap.com/)
(getbootstrap.com).
- Agrega las dependencias de JavaScript a tu proyecto de ser necesarias.