**Diseño y Desarrollo**

**De Una**

**Plataforma Web Para Recetas**

**ICI 324 Bases de Datos y Programación Web**

**27/09/24**

**integrantes**

Matias Carrasco (matias.carrasco@alumnos.uv.cl)

Graciela Suárez (graciela.suarez@alumnos.uv.cl)

Joyce Vivar (joyce.vivar@alumnos.uv.cl)

**Profesor**

Benjamin Serrano Barba

1. Introducción

En la vida de los estudiantes que durante la temporada universitaria deben independizarse buscar recetas nuevas, pedir comida en lugares de comida rápida o incluso cocinar los mismos platos repetidamente resulta monótono y aburrido, es por esto que buscan nuevas recetas que les permitan variar sus comidas, pero encontrar recetas nuevas y satisfactorias que se ajusten a habilidad y tiempo es difícil.

Se propone la creación de un sitio web que con el objetivo de facilitar esta búsqueda, se dedique a la recopilación y administración de recetas de cocina. Este sitio permitirá que cada usuario pueda crear una cuenta personal, explorar recetas en base a su dificultad y tiempo de preparación, guardar sus recetas favoritas e incluso modificarlas.

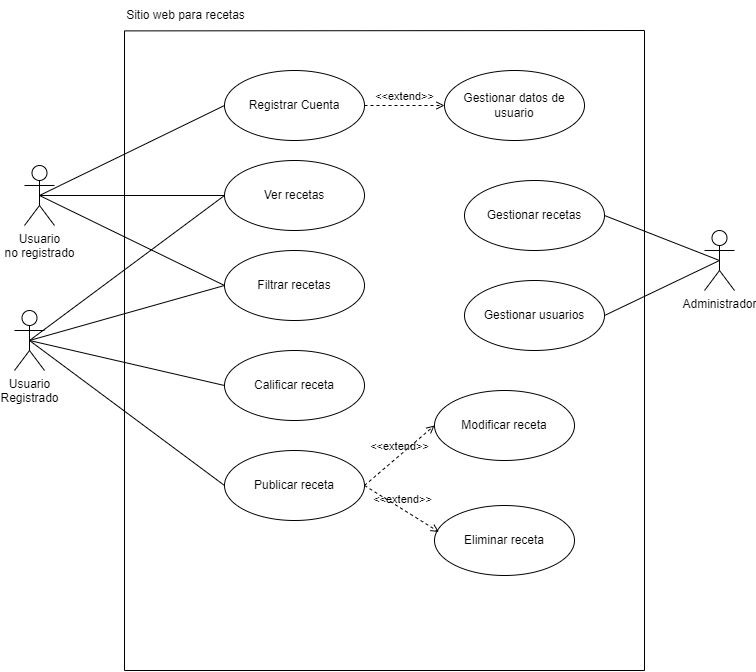
Cada receta seguirá un formato por defecto, lo que para que sea fácil de leer y modificar para todos los usuarios, el formato contará con un nombre, autor, dificultad, pasos, ingredientes necesarios, tiempo estimado de preparación y un espacio para colocar un link que lleve a un video tutorial de la receta o para subir una imagen del resultado final.

Este sitio estará diseñado no solo como herramienta útil para estudiantes, sino también para cualquier usuario que esté interesado en compartir, encontrar, guardar o preparar nuevas recetas con facilidad, ya sean novatos o expertos.

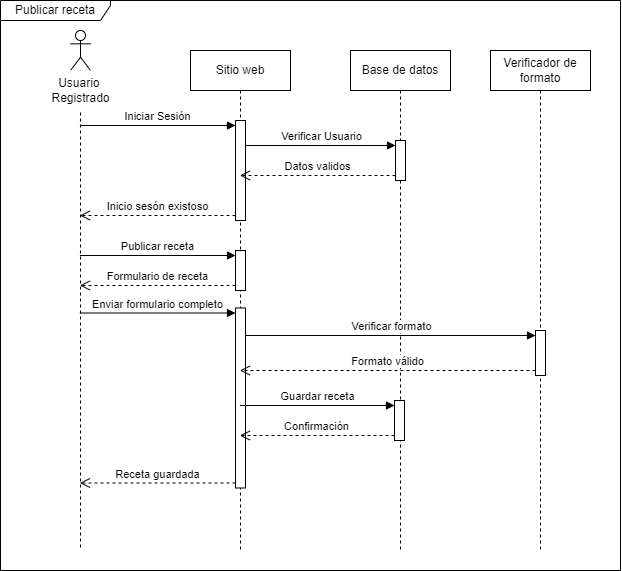
Para almacenar la información de los usuarios y las recetas se utilizará la base de datos MongoDB y para la creación de la plataforma web se usará el lenguaje de programación PHP para la lógica del sistema junto a herramientas como Bootstrap para el diseño del sitio.

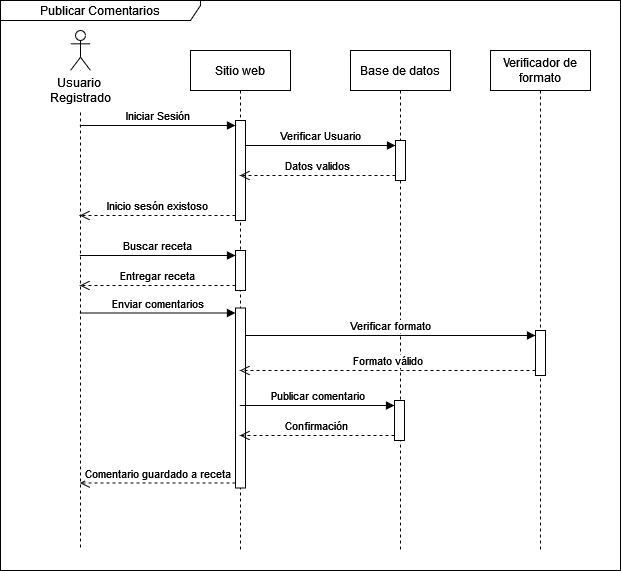
1. Requerimientos
   1. Requerimientos funcionales

* Los usuarios deben poder registrarse en el sitio, proporcionando información necesaria para su cuenta, como nombre y correo electrónico.
* Los usuarios pueden modificar sus datos de usuario.
* Aquellos usuarios que no estén registrados, sólo pueden visualizar las recetas que ya se encuentran en el sitio.
* Los usuarios deben estar registrados para poder añadir nuevas recetas al sitio, al igual que para añadir a su colección.
* Los usuarios registrados pueden crear, modificar o eliminar recetas creadas por ellos mismos o de su colección (favoritas).
  1. Requerimientos no funcionales
* La búsqueda en el sitio debe ser simple y rápida, permitiendo a los usuarios encontrar las recetas de manera eficiente.2
* Es necesario proporcionar una dirección de correo electrónico válida para poder crear una cuenta de usuario en el sitio.
  1. Diagrama de casos de uso

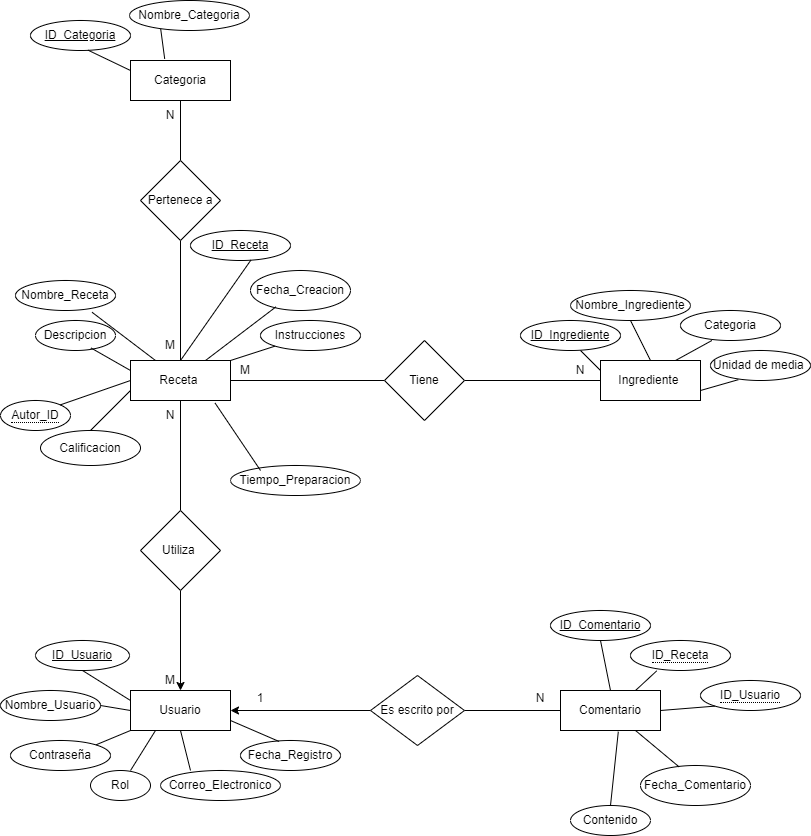


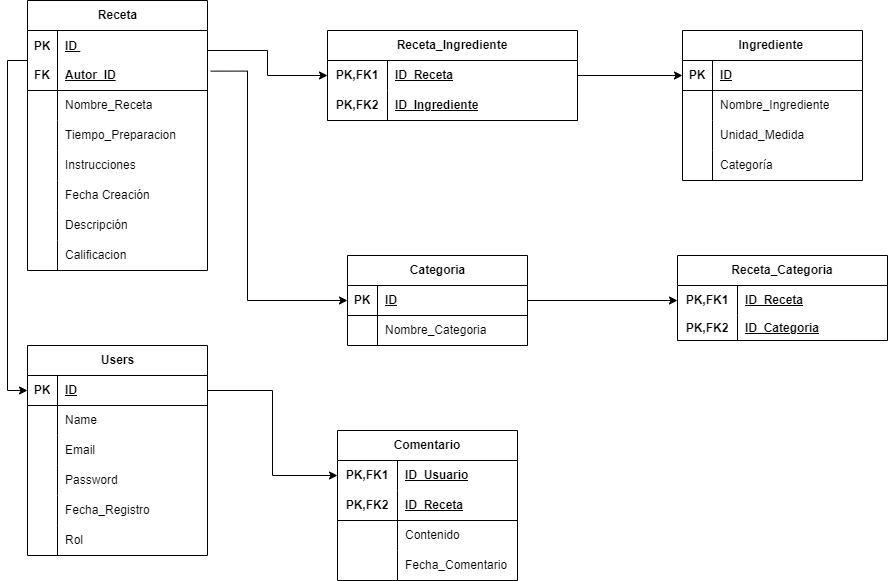
* 1. Diagrama de secuencia de la solución





1. Modelo de datos
   1. Modelo Entidad Relación y relacional





* 1. Diccionario de datos

| **Nombre Entidad** | **Nombre Atributo** | **PK/FK** | **Tipo de Dato** | **NULL/NOT NULL** | **Valores**  **por**  **Defect**o |
| --- | --- | --- | --- | --- | --- |
| Recetas | id  nombre\_receta  descripción  tiempo\_preparacion  instrucciones  fecha\_creacion  autor\_id  calificación | PK  FK | Bigint unsigned  VARCHAR (50)  VARCHAR (100)  TIME  VARCHAR (500)  timestamp  Bigint unsigned  INT | NOT NULL  NULL  NULL  NULL  NULL  NOT NULL  NOT NULL  NOT NULL | ’  ‘Ensalada Mediterránea’  ‘00:00:00’  fecha y hora actual  calificación del 0 al 5 |
| Ingredientes | id  nombre\_Ingrediente  unidad\_Medida  categoría | PK | Bigint unsigned  VARCHAR(50)  VARCHAR(20)  VARCHAR(20) | NOT NULL  NOT NULL  NULL  NOT NULL | ‘harina’  ‘Pastas’ |
| receta\_\_ingredientes | ID\_receta  ID\_Ingrediente | FK  FK | Bigint unsigned  Bigint unsigned | NOT NULL  NOT NULL |  |
| Users | id  name  email  password  fecha\_registro  rol | PK | Bigint unsigned  VARCHAR(255)  VARCHAR (255)  VARCHAR (255)  TIMESTAMP  VARCHAR(20) | NOT NULL  NOT NULL  NOT NULL  NOT NULL  NOT NULL  NULL | ‘Daniel perez’  ‘no\_disponible@example.com’  \*\*\*\*\*\*  ‘(fecha y hora actual)  rol : ‘chef’,’admin’,’usuario normal’ |
| Comentarios | id  contenido  Fecha\_comentario  ID\_Usuario  ID\_Receta | FK  FK | Bigint unsigned  VARCHAR (90)  TIMESTAMP  Bigint unsigned  Bigint unsigned | NOT NULL  NOT NULL  NOT NULL  NOT NULL  NOT NULL |  |
| Categorías | id  nombre\_categoria | PK | Bigint unsigned  VARCHAR(20) | NOT NULL  NOT NULL |  |
| receta\_\_categorias | id  ID\_Receta  ID\_Categoria | PK  FK  FK | Bigint unsigned  Bigint unsigned  Bigint unsigned | NOT NULL  NOT NULL  NOT NULL |  |

* 1. Integridad referencial y supuestos

La integridad referencial en nuestra base de datos intenta asegurar coherencia en las relaciones entre tablas como recetas , ingredientes , usuarios y comentarios . Establece asimismo reglas para las operaciones de eliminación y actualización ;para evitar datos huérfanos y mantener la consistencia , como se muestra a continuación :

1. Tabla Receta:

Autor\_ID (FK) referencia a Usuario(ID\_Usuario) ON DELETE: SET NULL ON UPDATE: CASCADE Supuesto: Si se elimina un usuario, sus recetas se mantienen pero sin autor asignado.

1. Tabla Receta Ingrediente:

ID\_Receta (FK) referencia a Receta(ID\_Receta) ON DELETE: CASCADE ON UPDATE: CASCADE

ID\_Ingrediente (FK) referencia a Ingrediente(ID\_Ingrediente) ON DELETE: RESTRICT ON UPDATE: CASCADE Supuesto: Si se elimina una receta, se eliminan sus ingredientes asociados. No se puede eliminar un ingrediente si está en uso en alguna receta.

1. Tabla Comentario:

Usuario\_ID (FK) referencia a Usuario(ID\_Usuario) ON DELETE: CASCADE ON UPDATE: CASCADE

Receta\_ID (FK) referencia a Receta(ID\_Receta) ON DELETE: CASCADE ON UPDATE: CASCADE Supuesto: Si se elimina un usuario o una receta, se eliminan todos los comentarios asociados.

1. Tabla Receta Categoría:

ID\_Receta (FK) referencia a Receta(ID\_Receta) ON DELETE: CASCADE ON UPDATE: CASCADE

ID\_Categoria (FK) referencia a Categoria(ID\_Categoria) ON DELETE: RESTRICT ON UPDATE: CASCADE Supuesto: Si se elimina una receta, se eliminan sus asociaciones con categorías. No se puede eliminar una categoría si está asociada a alguna receta.

Los supuestos adicionales incluyen restricciones de unicidad y valores por defecto para garantizar la validez y relevancia de los datos almacenados.

1. Unicidad:
2. El campo Nombre\_Usuario en la tabla Usuario debe ser único.
3. El campo Correo Electrónico en la tabla Usuario debe ser único.
4. La combinación de ID\_Receta e ID Ingrediente en Receta\_Ingrediente debe ser única.
5. La combinación de ID\_Receta e ID\_Categoria en Receta\_Categoria debe ser única.
6. Valores por defecto:
7. Fecha\_Creacion en Receta: se establece automáticamente con la fecha actual al crear una nueva receta.
8. Fecha\_Registro en Usuario: se establece automáticamente con la fecha actual al crear un nuevo usuario.
9. Fecha\_Comentario en Comentario: se establece automáticamente con la fecha y hora actuales al crear un nuevo comentario.
10. Restricciones de dominio:
11. Calificación en Receta: debe ser un valor entre 0 y 5.
12. Cantidad en Receta\_Ingrediente: debe ser un valor positivo.
13. Rol en Usuario: debe ser uno de los siguientes valores: 'usuario', 'admin', 'chef'.
14. Otros supuestos:
15. Una receta debe tener al menos un ingrediente.
16. Una receta debe pertenecer al menos a una categoría.
17. El campo Recetas\_Internacional en Receta\_Categoria es un booleano (TRUE/FALSE) que indica si la receta es considerada internacional para esa categoría.
18. Gestión de eliminaciones:
19. No se permite la eliminación de ingredientes o categorías que estén en uso.
20. La eliminación de una receta implica la eliminación de todos sus datos asociados (ingredientes, categorías, comentarios).
21. Consultas

*Alter SQL*

***Consulta 1 :*** Añadir una columna a la tabla Receta para almacenar el número de porciones

ALTER TABLE Receta ADD COLUMN Porciones INT;

Álgebra Relacional:

***Consulta 2 :*** Modificar la columna Tiempo\_preparacion en la tabla Receta para incluir segundos, y añadir una columna para el tiempo de cocción

ALTER TABLE Receta MODIFY COLUMN Tiempo\_preparacion TIME(0),ADD COLUMN Tiempo\_coccion TIME(0) AFTER Tiempo\_preparacion;

Álgebra Relacional:

*Select SQL*

***Consulta 1:*** Obtener los ID(s) de los usuarios más activos junto con su rol en los últimos 30 días , basado en el número de recetas y comentarios publicados

SELECT u.ID\_Usuario, u.Rol, COUNT(DISTINCT r.ID\_receta) AS Total\_Recetas, COUNT(DISTINCT c.ID\_comentario) AS Total\_Comentarios FROM Usuario u LEFT JOIN Receta r ON u.ID\_Usuario = r.Autor\_id AND r.Fecha\_creacion >= (SELECT MIN(Fecha) FROM Tabla\_Fechas WHERE Condicion = 'últimos\_30\_días') LEFT JOIN Comentario c ON u.ID\_Usuario = c.Usuario\_ID AND c.Fecha\_comentario >= (SELECT MIN(Fecha) FROM Tabla\_Fechas WHERE Condicion = 'últimos\_30\_días') GROUP BY u.ID\_Usuario, u.Rol ORDER BY (Total\_Recetas + Total\_Comentarios) DESC;

Álgebra Relacional :

τ (Total\_Recetas + Total\_Comentarios) DESC (γ u.ID\_Usuario, u.Rol; COUNT(DISTINCT r.ID\_receta)->Total\_Recetas, COUNT(DISTINCT c.ID\_comentario)->Total\_Comentarios (

π u.ID\_Usuario, u.Rol, r.ID\_receta, c.ID\_comentario (Usuario u ⟕ (σ r.Fecha\_creacion >= (π MIN(Fecha) (σ Condicion = 'últimos\_30\_días' (Tabla\_Fechas))) (Receta r)) ON u.ID\_Usuario = r.Autor\_id ⟕ (σ c.Fecha\_comentario >= (π MIN(Fecha) (σ Condicion = 'últimos\_30\_días' (Tabla\_Fechas))) (Comentario c)) ON u.ID\_Usuario = c.Usuario\_ID)))

***Consulta 2* :** Calcular el Promedio de la Calificación de las recetas de cada Usuario y luego seleccionar al Usuario con el Promedio más alto

SELECT u.Nombre\_Usuario , AVG() AS Promedio\_Calificacion FROM Usuario u JOIN Receta r ON u.ID\_Usuario = r.Autor\_ID GROUP BY u.ID\_Usuario, u,Nombre\_Usuario ORDER BY Promedio\_Calificacion DESC LIMIT 1;

Álgebra Relacional:

Nombre\_Usuario, Promedio\_Calificacion ( Promedio\_Calificacion DESC ( u.ID\_Usuario, u.Nombre\_Usuario; AVG(r.Calificacion)Promedio\_Calificacion (Usuario u ⋈ u.ID\_Usuario = r.Autor\_ID Receta r))) [1]

***Consulta 3:*** Mostrar todos los datos de los comentarios para las recetas de tipo “postre ”, y mostrar también el estado del comentario

SELECT c.\*FROM comentario c JOIN Receta r ON c.Receta\_ID = r.ID\_Receta JOIN Receta\_Categoria rc ON r.ID\_Receta =rc.Receta \_ID JOIN Categoria cat ON rc.Categoria\_ID = cat.ID\_Categoria WHERE cat.NOmbre\_Categoria =’POSTRE’

Álgebra Relacional:

c.\* ( cat.Nombre\_Categoria ='POSTRE'(comentario c⋈c.Receta\_ID = r.ID\_Receta Receta r⋈ r.ID\_Receta = rc.Receta\_ID Receta\_Categoria rc ⋈ rc.Categoria\_ID = cat.ID\_Categoria Categoria cat))

Update SQL

***Consulta 1:*** Actualizar el tiempo de preparación de una receta

UPDATE Receta SET tiempo\_preparacion = ‘00:45:00’ WHERE ID\_receta = ‘REC001’

Algebra Relacional :

***Consulta2 :*** Actualizar el Rol del usuario

UPDATE Usuario SET Rol = ‘Chef’ WHERE ID\_Usuario = ’111111-1’

Algebra Relacional:

INSERT SQL

***Consulta 1:*** Insertar un nuevo usuario

INSERT INTO Usuario (ID\_Usuario, Nombre\_Usuario, correo\_electronico, Contraseña, Fecha\_registro, Rol) VALUES ('222222-2', 'Maria Lopez', 'maria.lopez@example.com', 'contraseña123', CURRENT\_DATE, 'usuario normal');

Álgebra Relacional:

***Consulta 2:*** Insertar una nueva receta

INSERT INTO Receta (ID\_receta, Nombre\_receta, Descripción, Tiempo\_preparacion, Instrucciones, Fecha\_creacion, ID\_Autor, Calificación) VALUES ('REC002', 'Pasta Carbonara', 'Deliciosa pasta italiana', '00:30:00', 'Cocinar la pasta, preparar la salsa...', CURRENT\_DATE, '222222-2', 4);

Álgebra Relacional:

***Consulta 3:*** Insertar un nuevo ingrediente

INSERT INTO Ingrediente (ID\_Ingrediente, Nombre\_Ingrediente, Unidad\_Medida, Categoría) VALUES ('ING002', 'huevos', 'unidad', 'Lácteos');

Álgebra Relacional:

DELETE SQL

***Consulta 1:*** Eliminar un comentario específico

DELETE FROM Comentario WHERE ID\_Usuario = '111111-1' AND ID\_Receta = 'REC001';

Álgebra Relacional:

***Consulta 2:*** Eliminar un ingrediente de una receta

DELETE FROM Receta\_Ingrediente WHERE ID\_Receta = 'REC001' AND ID\_Ingrediente = 'ING001';

Álgebra Relacional:

DROP SQL

***Consulta 1:*** Eliminar la tabla Categoría

DROP TABLE Categoría; = 'ING001';

La sentencia SQL 'DROP TABLE Categoría;' no tiene un equivalente en álgebra relacional, ya que el álgebra relacional no incluye operaciones para eliminar estructuras de tablas.

1. Actualización de Datos

En esta entrega se a agregado un nuevo diagrama de secuencia , correspondiente a la fase de la publicación de comentarios que tiene el usuario en el sitio web , de la misma manera se realizaron cambios en algunas de las variables del diccionario de datos , ajustándose tanto algunos nombres como tipos de datos para reflejar mejor los requerimientos del sistema y facilitar su implementación.

Estos cambios fueron necesarios para alinear los datos con las nuevas funcionalidad esa agregadas , por otro lado el modelo relacional ha sido actualizado en esta entrega incorporando algunos cambios introducidos en el diccionario de datos , como la sugerencia mencionada en el feedback de la entrega anterior.