

# RecipeBoxCloud

# Azure & AWS

---



**PONDERACIÓN: 15**

**Horas Aproximadas: 15**

Universidad San Carlos de Guatemala

Facultad de ingeniería.

Ingeniería en ciencias y sistemas

# Índice

|  |           |
|--|-----------|
| <b>Índice .....</b>                    | <b>1</b>  |
| <b>Competencias.....</b>               | <b>2</b>  |
| <b>Objetivos .....</b>                 | <b>2</b>  |
| General.....                           | 2         |
| Específicos .....                      | 2         |
| <b>Descripción / Enunciado.....</b>    | <b>2</b>  |
| Servicios .....                        | 3         |
| Funcionalidades .....                  | 3         |
| Registro de usuarios.....              | 3         |
| Inicio de Sesión .....                 | 3         |
| Explorar recetas .....                 | 3         |
| Guardar recetas .....                  | 4         |
| Recetas Favoritas .....                | 4         |
| Crear y Compartir recetas .....        | 4         |
| Mis recetas.....                       | 4         |
| Implementación.....                    | 4         |
| Arquitectura .....                     | 4         |
| Web estática.....                      | 5         |
| Almacenamiento de fotos de perfil..... | 5         |
| Serverless .....                       | 5         |
| Base de Datos.....                     | 5         |
| Servidores .....                       | 6         |
| Balanceador de carga .....             | 6         |
| Manual Técnico .....                   | 6         |
| <b>Entregables.....</b>                | <b>6</b>  |
| <b>Consideraciones.....</b>            | <b>7</b>  |
| <b>Cronograma.....</b>                 | <b>7</b>  |
| <b>Rúbrica de Calificación .....</b>   | <b>7</b>  |
| <b>Valores .....</b>                   | <b>10</b> |
| <b>Referencias .....</b>               | <b>11</b> |

## Competencias

- Explica el concepto de Cloud Computing, sus modelos de servicio y las ventajas y limitaciones de distintos proveedores Cloud mediante ejemplos cotidianos, casos de uso empresarial y comparativas para el desarrollo de infraestructuras escalables y rentables.
- Implementa servicios básicos de Cloud utilizando Amazon EC2, Amazon RDS, Amazon S3, Azure Virtual Machines, Azure SQL, Azure Blob Storage, Azure App Services y Azure Logic Apps para desplegar aplicaciones básicas, considerando la configuración de redes, seguridad y escalabilidad.
- Desarrolla aplicaciones serverless simples utilizando AWS Lambda y Azure Functions para procesar eventos y ejecutar la lógica de los negocios

## Objetivos

### General

Comprender y desarrollar una aplicación web con un proveedor de servicios en la nube nuevo con el fin de que los estudiantes adquieran experiencia práctica en el despliegue de soluciones Cloud, además de que comprendan las diferencias, similitudes y complejidades entre los proveedores vistos en el laboratorio.

### Específicos

- Desplegar una aplicación completamente en la nube
- Familiarizarse con el desarrollo Cloud con nuevos proveedores
- Comparar la complejidad de la implementación de servicios de Azure

## Descripción / Enunciado

Microsoft Azure es un proveedor de servicios de nube que ofrece una gran cantidad de servicios de infraestructura que ayudan a las empresas a escalar y crecer, con esto en mente se desarrollará una aplicación que permita a los usuarios crear, guardar, compartir y explorar recetas de cocina.

Para lograr esto los estudiantes implementarán distintos servicios de Azure que son listados a continuación.

## Servicios

- Azure VM
- Azure Load Balancer
- Azure Blob Storage
- Azure App Functions
- Azure Api Management
- AWS RDS o DynamoDB

## Funcionalidades

### Registro de usuarios

Para el registro de usuarios nuevos se deberá llenar un formulario donde se le requerirán a los usuarios los siguientes recursos:

- Nombre de usuario
- Correo electrónico
- Contraseña
- Confirmación de la contraseña
- Imagen de perfil

### Consideraciones:

- *El nombre de usuario debe ser único en la plataforma, no pueden existir 2 usuarios con el mismo nombre de usuario.*
- *Las contraseñas deben ser encriptadas antes de almacenarse en la base de datos.*

### Inicio de Sesión

Todo usuario podrá iniciar sesión mediante las siguientes credenciales:

- Nombre de Usuario
- Contraseña

### Explorar recetas

Al iniciar sesión el usuario deberá poder ver todas las recetas publicadas por los demás usuarios en la plataforma, la forma de desplegarlas queda a discreción del estudiante.

## Guardar recetas

En la sección principal el usuario debe ser capaz de guardar cualquiera de ellas mediante un botón de “Favoritos” para poder verlas a futuro.

## Recetas Favoritas

En esta sección el usuario deberá poder ver las recetas que guarde como favoritas de la página principal.

## Crear y Compartir recetas

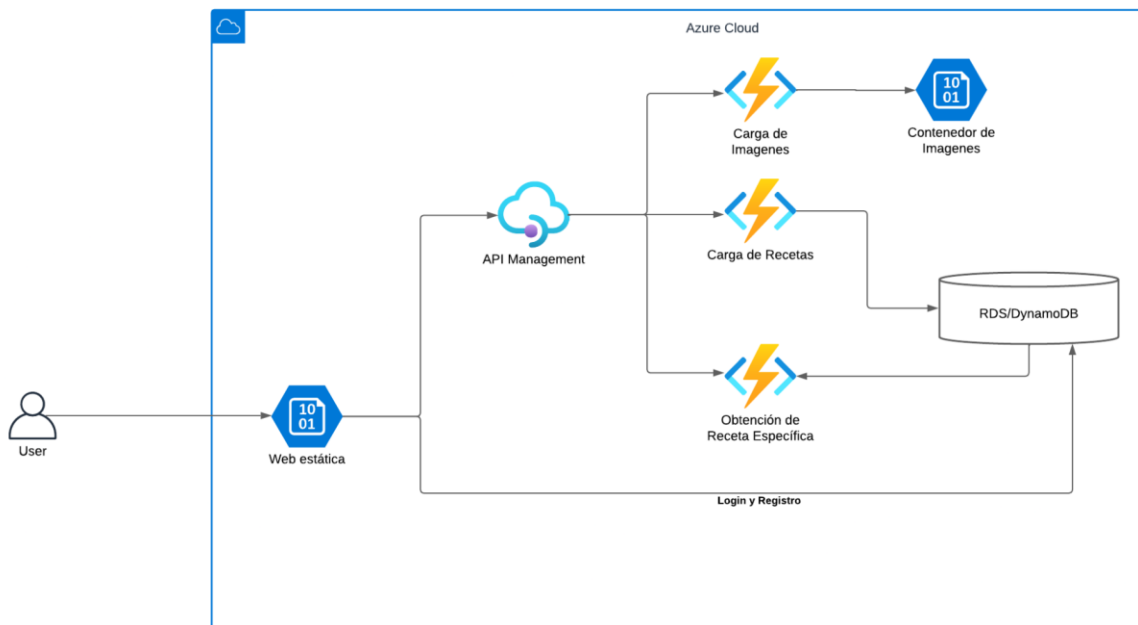
Cualquier usuario debe ser capaz de crear una receta, al terminar de redactar dicha receta debe ser posible guardarla y de manera automática será compartida en toda la plataforma.

## Mis recetas

El usuario deberá ser capaz de entrar a esta sección para ver cada una de las recetas que él mismo crea y comparte.

## Implementación

### Arquitectura



## Web estática

Para la aplicación web estática, se requerirá de un cliente desarrollado en la tecnología seleccionada por el estudiante, esta deberá ser compilada y al obtener los archivos estáticos deberán de ser cargadas a **un Blob Container de Azure** para que se pueda acceder a cualquier momento y pueda conectarse al balanceador de carga.

El nombre de la cuenta de almacenamiento y el container queda a discreción del estudiante, pero debe identificarse cómo la web estática.

## Almacenamiento de fotos de perfil

Para alojar las imágenes del perfil del usuario es necesario usar un **Blob de Blob Storage** de Azure, el nombre queda a decisión del estudiante siempre y cuando identifique que es el almacenamiento de dichas imágenes.

Es importante asegurar que todas las fotos sean públicas, para que se puedan acceder desde la aplicación mediante la dirección URL del objeto.

## Serverless

Con el objetivo de entender e implementar las arquitecturas Serverless será necesario implementar tanto **Azure App Functions como Azure Api Management**.

En esta práctica se solicita que se implementen 3 funciones obligatorias:

- Una función para cargar las fotos de perfil a Blob Storage.
- Una función para cargar las recetas a la base de datos.
- Una función para obtener los datos de una receta en específico.

Cada una de estas funciones deberá ser vinculada a una API con Api Management para su uso dentro de la aplicación.

## Base de Datos

Para la base de datos de la solución se requerirá el uso del servicio **Amazon RDS o Amazon DynamoDB** y se deberá tomar las siguientes consideraciones:

- La base de datos deberá contener toda la información necesaria para almacenar la lógica de negocio de la solución.
- La contraseña del usuario tiene que estar encriptada
- **NO se deberán almacenar las imágenes directamente en la base de datos**, para ello se recomienda almacenar únicamente la URL del objeto de las imágenes almacenadas.

## Servidores

Estos componen el Back-End de la solución, se deberá tomar en cuenta las siguientes consideraciones:

1. Se deberán realizar 2 servidores con exactamente las mismas funciones, pueden o no usar lenguajes distintos.
2. Cada uno de los servidores deberá estar montado en una instancia de **VM de Azure**.
3. Se deberá configurar el grupo de seguridad de ambas instancias únicamente con los puertos que necesite su servidor.

## Balanceador de carga

Para balancear la carga entre las instancias de VM de Azure **se utilizará el balanceador respectivo del proveedor**, se deberá tomar en cuenta las siguientes consideraciones:

1. El balanceador debe permitir que la aplicación web pueda seguir en funcionamiento después de apagar uno de los 2 servidores
2. El balanceador debe estar configurado para redirigir el tráfico a las 2 instancias que poseen los diferentes servidores.

## Manual Técnico

Se requiere que se realice un manual redactado en formato Markdown dentro del repositorio, lo que se debe adjuntar al manual es lo siguiente:

1. Datos de los estudiantes.
2. Descripción de la arquitectura que utilizaron.
3. Capturas de pantalla de los recursos:
  - a. Blob Containers de Azure
  - b. Instancias de VM de Azure
  - c. Balanceador de carga de Azure
  - d. Funciones de Azure Functions
  - e. Configuración de Api Management
4. Una conclusión sobre las diferencias percibidas entre el desarrollo de Azure y AWS

## Entregables

El entregable será la URL de su repositorio GitHub en UEDI.

## Consideraciones

- Nombre del repositorio: Semi1-Sección-2S2025-Grupo#-Practica2 (Donde # es su número de grupo).
- Repositorio de Github debe estar en modo privado y documentado con el formato Markdown (manual de configuraciones)
- La página web deberá estar cargada en Blob Storage.
- NO se aceptará ni calificará nada de forma local, todo deberá estar desplegado en la nube de AWS.
- Agregar como colaborador en el repositorio al usuario del auxiliar del laboratorio dependiendo de la sección.
  - Auxiliar 1: Emerson-O
  - Auxiliar 2: CamiloSincal
- El proyecto debe desarrollarse en grupos.

## Cronograma

| Tarea   | Fecha                 |
|---|-----------------------|
| Asignación de la práctica / Entrega del enunciado | Septiembre 02 de 2025 |
| Fecha de entrega                                  | Septiembre 16 de 2025 |
| Fecha de calificación                             | Septiembre 19 de 2025 |

## Rúbrica de Calificación

| Criterio                                 | Descripción   | Puntos Máximos |
|--|---|----------------|
| <b>Configuración de la Base de Datos</b> | Se crea una base de datos no relacional o relacional alojada en la nube que cubre todas las necesidades de la aplicación. | 3              |



|   |   |   |
|---|---|---|
| <b>Contenedor del Sitio Web estático</b>      | El sitio web se despliega y se consume desde la web estática de Azure Blob Storage.   | 5 |
| <b>Contenedor de imágenes</b>                 | Se crea un Contenedor para almacenar las imágenes o fotos de perfil de los usuarios   | 5 |
| <b>Instancia del Servidor 1</b>               | Se crea una máquina virtual con EC2 para almacenar la API 1   | 5 |
| <b>Instancia del Servidor 2</b>               | Se crea una máquina virtual con EC2 para almacenar la API 2   | 5 |
| <b>Grupos de Seguridad de las instancias</b>  | Ambas instancias de Azure VM cuentan con los grupos de seguridad debidamente configurados para restringir el acceso a los puertos necesarios. | 3 |
| <b>Configuración del Balanceador de Carga</b> | Se crea el balanceador de carga y ambas instancias de VM se encuentran asignadas a esta.  | 3 |
| <b>Tráfico entre redes</b>                    | Al detener una instancia de una VM la aplicación sigue funcionando sin ningún problema.   | 6 |
| <b>Función de almacenamiento de fotos</b>     | Se cuenta con la función de Azure App Function que permite cargar las imágenes a Blob Storage.  | 5 |
| <b>Función de guardado de Recetas</b>         | Se cuenta con la función de Azure App Function que permite cargar las recetas a la base de datos de AWS.                                      | 5 |
| <b>Función de Búsqueda de Recetas</b>         | Se cuenta con la función de Azure App Function que permite recuperar los datos de una receta de la base de datos.                             | 5 |
| <b>Configuración de Azure API Management</b>  | Todas las funciones se encuentran debidamente configurados dentro de  | 3 |

|                                     |  |   |
|-------------------------------------|--|---|
|                                     | la API de Azure y en rutas diferentes.   |   |
| <b>Implementación de la API</b>     | La API de Azure es consumida por la web estática en Blob.  | 5 |
| <b>Iniciar Sesión</b>               | Cualquier usuario registrado puede ingresar a la página web con sus credenciales correctas.                            | 3 |
| <b>Registrar Usuario</b>            | Cualquier persona es capaz de registrarse en la aplicación siempre y cuando se cumplan los requisitos de comprobación. | 3 |
| <b>Explorar Recetas</b>             | Cualquier usuario que inicie sesión es capaz de visualizar las diferentes recetas disponibles en la página web.        | 3 |
| <b>Crear Recetas</b>                | Cualquier usuario es capaz de escribir y guardar su propia receta en el sistema.                                       | 5 |
| <b>Compartir Recetas</b>            | Al guardar una receta esta se muestra debidamente en la sección principal del sistema.                                 | 5 |
| <b>Guardar Receta</b>               | El usuario es capaz de guardar en “Mis Favoritos” cualquier receta disponible en la página principal.                  | 3 |
| <b>Sección de Recetas Favoritas</b> | Los usuarios pueden ver cualquier receta y su información completa que hayan guardado en la sección de mis favoritos.  | 3 |
| <b>Sección de “Mis Recetas”</b>     | Se puede acceder a la sección “Mis Recetas” para observar todas las recetas escritas por el usuario.                   | 4 |
| <b>Manual Técnico</b>               | El manual técnico cuenta con la información de los estudiantes, la lista de configuraciones de los                     | 3 |

|                                  |  |     |
|----------------------------------|--|-----|
|                                  | servicios, la arquitectura y la conclusión comparativa de proveedores.                               |     |
| <b>Preguntas de conocimiento</b> | Los estudiantes responden de manera adecuada a las preguntas relacionadas al laboratorio y práctica. | 10  |
| <b>TOTAL</b>                     |  | 100 |

## Valores

En el desarrollo de la práctica, se espera que cada estudiante demuestre honestidad académica y profesionalismo. Por lo tanto, se establecen los siguientes principios:

### 1. Originalidad del Trabajo

- Cada estudiante o equipo debe desarrollar su propio código y/o documentación, aplicando los conocimientos adquiridos en el curso.

### 2. Prohibición de Copias y Plagio

- Si se detecta la copia total o parcial del código, documentación o cualquier otro entregable, la calificación será de **0 puntos**.
- Esto incluye la reproducción de código entre compañeros, la reutilización de proyectos de semestres anteriores o el uso de código externo sin la debida referencia.

### 3. Uso Responsable de Recursos Externos

- El uso de bibliotecas, frameworks y ejemplos de código externos está permitido, siempre y cuando se referencian correctamente y se comprendan plenamente. (Consultar con el catedrático su política)

### 4. Honestidad en el Trabajo en Grupo

- Si un estudiante no puede presentarse a la calificación y no puede demostrar su participación en el desarrollo de la práctica **se le aplicará una penalización del 30% sobre la nota total obtenida en grupo**.
- Si un integrante del grupo no participa en ningún aspecto de la práctica, su nota individual será automáticamente de **0 puntos**.

### 5. Revisión y Detección de Plagio

- Se podrán utilizar herramientas automatizadas y revisiones manuales para identificar similitudes en los proyectos.
- En caso de sospecha, el estudiante deberá justificar su código y demostrar su desarrollo individual o en equipo. Si este extremo no es comprobable la calificación será de **0 puntos**.

Al detectarse estos aspectos se informará al catedrático del curso quien realizará las acciones que considere oportunas.

## Referencias

1. Microsoft Azure. (s. f.). Documentación de servicios de Azure.  
<https://azure.microsoft.com/es-es/services/>