**Semana 1 - Taller 3.1**

**SaaS con Azure y AWS: Desarrollo de una Aplicación de Software como Servicio (SaaS)**

1. **Objetivo:**

En este taller, los estudiantes aprenderán a desarrollar una aplicación de Software como Servicio (SaaS) utilizando servicios específicos de Microsoft Azure y Amazon Web Services (AWS). Se centrarán en la creación de una aplicación web escalable y multiusuario.

1. **Recursos necesarios:**

Cuentas registradas en Microsoft Azure y AWS.

Conocimientos básicos de desarrollo web y conceptos generales de SaaS.

Familiaridad con lenguajes de programación como Node.js, Python o C#.

1. **Taller de tipo:**

Grupal o individual, siempre y cuando se encuentren presentes en el laboratorio.

1. **Pasos a realizar:**

Se recomienda tomar capturas del procedimiento para su informe.

**Parte 1: Implementación en Microsoft Azure (viernes)**

**Paso 1: Creaci√≥n de una Base de Datos SQL en Azure SQL Database**

* Acceder al portal de Azure.
* Navegar a "Crear un recurso" y seleccionar "Bases de datos + Cach√©" y luego "Base de datos SQL".
* Configurar la base de datos con un nombre √∫nico, configuraci√≥n del servidor y otros detalles.
* Crear una tabla "Usuarios" con los siguientes campos:

Id (int, clave primaria, autoincremental)

Nombre (nvarchar(50))

CorreoElectronico (nvarchar(100))

Contrase√±a (nvarchar(100))

**Paso 2: Desarrollo de la Aplicaci√≥n Web en Azure**

* Crear una aplicación web utilizando Azure App Service.
* Utilizar un marco de desarrollo como ASP.NET (C#) o Node.js con Express (Node.js).
* Conectar la aplicaci√≥n web a la base de datos SQL mediante la cadena de conexión.
* Implementar funciones de registro, inicio de sesi√≥n y gesti√≥n de usuarios en la aplicación web.

**Ejemplo de Modelo de Usuario en C# (ASP.NET):**

public class Usuario

{

public int Id { get; set; }

public string Nombre { get; set; }

public string CorreoElectronico { get; set; }

public string Contraseña { get; set; }

}

**Ejemplo de Ruta de Registro en ASP.NET MVC (C#):**

[HttpPost]

public ActionResult Registro(Usuario nuevoUsuario)

{

// Validar y agregar el nuevo usuario a la base de datos

// (Reemplazar esto con la lógica real, incluyendo el cifrado de contraseña).

// Ejemplo de consulta SQL (Entity Framework):

// dbContext.Usuarios.Add(nuevoUsuario);

// dbContext.SaveChanges();

return RedirectToAction("InicioSesion");

}

**Ejemplo de Ruta de Inicio de Sesión en ASP.NET MVC (C#):**

[HttpPost]

public ActionResult InicioSesion(string correo, string contraseña)

{

// Validar las credenciales del usuario

// (Reemplazar esto con la lógica real, incluyendo la comparación de contraseñas).

// Ejemplo de consulta SQL (Entity Framework):

// var usuario = dbContext.Usuarios.FirstOrDefault(u => u.CorreoElectronico == correo && u.Contraseña == contraseña);

if (usuario != null)

{

// Usuario autenticado, realizar acciones necesarias (por ejemplo, establecer cookies de sesión).

return RedirectToAction("Inicio");

}

else

{

// Credenciales incorrectas, manejar el caso de error.

return View("InicioSesion");

}

}

**Parte 2: Implementación en Amazon Web Services (AWS) (sábado)**

**Paso 4: Creación de una Base de Datos en AWS RDS (Relational Database Service)**

* Acceder al panel de AWS.
* Navegar a "Servicios" y seleccionar "Bases de datos" y luego "RDS".
* Crear una instancia de base de datos RDS para MySQL, PostgreSQL o el motor de tu elección.
* Configurar la base de datos con detalles como nombre de la instancia, nombre de usuario y contraseña.

**Paso 5: Desarrollo de la Aplicación Web en AWS**

* Crear una aplicación web utilizando AWS Elastic Beanstalk u otro servicio de aplicación web.
* Utilizar un marco de desarrollo como Flask (Python) o Express (Node.js).
* Conectar la aplicación web a la base de datos RDS mediante la cadena de conexión.
* Implementar funciones de registro, inicio de sesión y gestión de usuarios en la aplicación web.

**Ejemplo de Modelo de Usuario en Python (Flask):**

class Usuario(db.Model):

id = db.Column(db.Integer, primary\_key=True)

nombre = db.Column(db.String(50))

correo\_electronico = db.Column(db.String(100))

contraseña = db.Column(db.String(100))

**Ejemplo de Ruta de Registro en Flask (Python):**

@app.route('/registro', methods=['POST'])

def registro():

# Validar y agregar el nuevo usuario a la base de datos

# (Reemplazar esto con la lógica real, incluyendo el cifrado de contraseña).

# Ejemplo de consulta SQL (SQLAlchemy):

# nuevo\_usuario = Usuario(nombre=request.form['nombre'], correo\_electronico=request.form['correo'], contraseña=request.form['contraseña'])

# db.session.add(nuevo\_usuario)

# db.session.commit()

return redirect(url\_for('inicio\_sesion'))

**Ejemplo de Ruta de Inicio de Sesión en Flask (Python):**

@app.route('/inicio\_sesion', methods=['POST'])

def inicio\_sesion():

correo = request.form['correo']

contraseña = request.form['contraseña']

# Validar las credenciales del usuario

# (Reemplazar esto con la lógica real, incluyendo la comparación de contraseñas).

# Ejemplo de consulta SQL (SQLAlchemy):

# usuario = Usuario.query.filter\_by(correo\_electronico=correo, contraseña=contraseña).first()

if usuario:

# Usuario autenticado, realizar acciones necesarias (por ejemplo, establecer cookies de sesión).

return redirect(url\_for('inicio'))

else:

# Credenciales incorrectas, manejar el caso de error.

return render\_template('inicio\_sesion.html')

1. **Presentación del taller**

En base a los “pasos a realizar” debe capturar el paso a paso de este procedimiento, para que luego sea integrado en su informe de clase.

Por tanto, deberá desarrollar en LATEX un informe sobre “**SaaS con Azure y AWS: Desarrollo de una Aplicación de Software como Servicio (SaaS)**”, con la siguiente estructura:

* Carátula
* Índice
* 1. Implementación en Microsoft Azure
* 2. Implementación en AWS
* 3. Conclusiones
* 4. Referencias

1. **Subir el resultado al aula virtual**

En el espacio correspondiente a la Actividad de la Semana, debe subir su informe en PDF y LATEX.

Atte.

Dr. Oscar J. Jimenez-Flores