

Business Case: Formularios de Captura de Documentos para Procesos de Crédito en AWS

Contexto:

En Creze, nuestro objetivo es facilitar los créditos para pymes en México mediante procesos eficientes y precisos. En este sentido, optimizar la captura de documentos del cliente es esencial. Para ello, necesitamos un desarrollador Fullstack que trabaje en la creación de un formulario de captura de documentos utilizando las capas gratuitas de AWS, así como tecnologías como AWS Lambda.

Objetivo del Caso:

Diseñar y desarrollar un formulario que permita a los usuarios cargar documentos necesarios para el proceso de crédito, fragmentar los archivos antes de la carga, enviarlos de manera segura y eficiente, y ensamblarlos en el servidor.

Componentes Claves del Business Case:

1. Descripción del Producto:

La aplicación permitirá a los clientes cargar documentos en fragmentos (En paralelo), con un tamaño máximo configurable. Estos documentos se enviarán usando protocolos HTTPS o REST a un backend que los ensamblará e indicará al front la carga exitosa.

2. Funcionalidades Esenciales:

- Login Seguro: Implementación de un sistema básico de autenticación.
- Formulario de Captura: Un formulario intuitivo que indique los documentos necesarios a cargar.
- Fragmentación y Carga: Fragmentación de documentos de máximo 1 mega (parametrizable) y carga monitoreada.
- Monitorización de Carga: Barra de progreso para mostrar el porcentaje de carga completada.
- Unión en Servidor: Ensamblado de los documentos una vez subidos al backend.

3. Proceso de Carga y Validación:

- Fragmentación Local de Archivos: Los documentos se dividirán localmente antes de subirlos.

- Carga en Fragmentos: Los fragmentos se enviarán usando HTTPS o REST.
- Monitorización del Progreso: Se deberá incluir un sistema de monitoreo del % de carga.
- Reensamblaje en AWS Lambda: Los fragmentos se juntarán en el backend de AWS.

4. Seguridad:

- Protocolos de Seguridad: Uso de HTTPS para la transferencia segura de datos.
- Autenticación: Implementación de un sistema de autenticación seguro.

5. Herramientas y Tecnologías:

- Backend: Libre selección de lenguajes permitidos en AWS Lambda.
- Frontend: React.js para crear una interfaz de usuario responsiva e intuitiva.
- Almacenamiento en la Nube: AWS S3 para almacenamiento de objetos.
- Procesamiento en la Nube: AWS Lambda para la lógica de reensamblaje de documentos.
- Monitoreo y Logs: CloudWatch para monitoreo y gestión de logs.
- Monitoreo y Logs: Tanto BDs, front y back deben ser servidos desde AWS Lambda usando las capas gratuitas de AWS.

Instrucciones:

Criterios de Evaluación:

- Claridad en la implementación: Visión clara del flujo de carga y ensamblado de documentos.
- Calidad del Código: Código limpio, eficiente y bien documentado.
- Seguridad: Implementación robusta y transferencia de datos.
- Eficiencia de Carga: Mecanismo eficiente de carga en fragmentos y ensamblaje en backend.
- Experiencia de Usuario: Interfaz intuitiva y muestra clara del progreso de carga.

Entregables:

1. Código de la Aplicación:

- Repositorio Git con el código backend y frontend.
- Scripts de configuración y deployment en AWS (incluyendo creación de recursos).
- Archivos de configuración y documentación técnica.

2. Documentación:

- Instrucciones detalladas para la configuración y ejecución del entorno de desarrollo y producción.
- Diagramas de arquitectura explicando el flujo de datos desde la carga hasta el ensamblaje.

3. Informe de Estrategia:

- Estrategia de autenticación y autorización de usuarios.
- Explicación del proceso de fragmentación, carga, y ensamblaje de documentos.
- Plan de pruebas y validación del sistema, incluyendo seguridad y manejo de errores.

Plazos:

- Fecha de entrega: Máximo 2 semanas luego de recibir el caso.
- Formato de entrega: Repositorio Git con documentación adjunta.
- Forma de envío: A través de la plataforma de aplicación o a `jobtech+developer@creze.com`.