

GEOVANNY TOVAR CEBALLOS

Tel. 3183774658

MANUAL APLICATIVO

PRUEBA 2024

Manual explicativo de la plataforma demo para las pruebas de selección

TABLA DE CONTENIDO

Contenido

Introducción	1
Diagrama de base de datos	2
BackEnd	3
FrontEnd	7
CI / CD	8
Diagrama de arquitectura	8

DOCUMENTO TECNICO DE APLICACIÓN PARA LA PRUEBA DE SELECCIÓN

Introducción

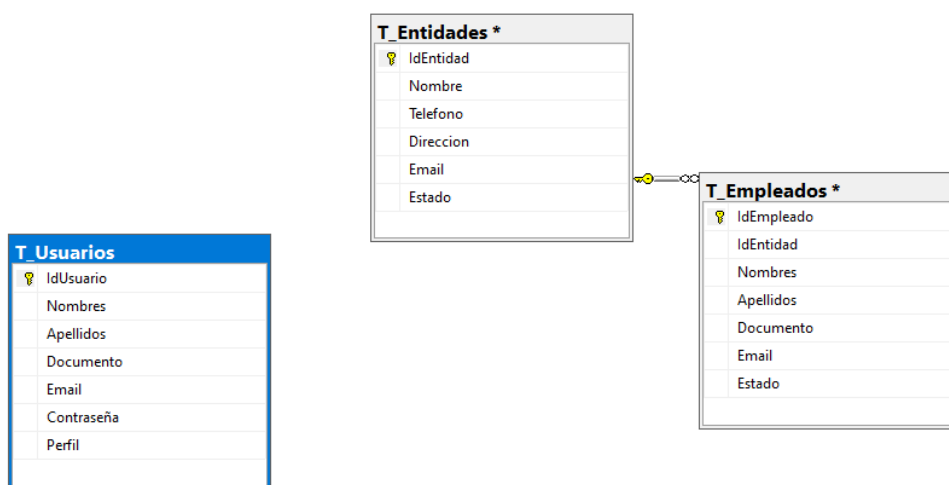
Este manual se ha creado con el fin de incorporar toda la arquitectura y solución planteada en la aplicación solicitada para las pruebas de selección.

Se ha desarrollado una aplicación web en .net con su respectivo backend y utilizando una base de datos con motor sql server en versión 20.2.30, de igual forma se ha dejado publicado el aplicativo en servidor cloud desplegado en IIS con el fin de tener operativa la plataforma y se han compartido los códigos en el Git indicado.

DOCUMENTO TECNICO DE APLICACIÓN PARA LA PRUEBA DE SELECCIÓN

Diagrama de base de datos

Hemos utilizado el motor SQL Server para almacenar la información de acuerdo a la solicitud realizada



Tres tablas las cuales son:

- Entidades
- Empleados
- Usuarios

En la tabla usuarios almacenamos los datos de inicio de sesión (login) de los que podrán acceder al aplicativo, en la tabla entidades almacenamos las entidades que pertenecen al negocio y en empleados se almacena la información de las personas que trabajan para cada entidad, es por esto que se crea una llave foránea para estas dos tablas que se comparten entre sí.

Debido a que nuestra arquitectura es basada en el modelo de micro servicios, hemos creado en la base de datos unos procedimientos que ejecutan el llamado de los servicios para retornar la información de la base de datos

- + **dbo.P_AdminActualizarEmpleados**
- + **dbo.P_AdminActualizarEntidades**
- + **dbo.P_AdminGenerarToken**
- + **dbo.P_AdminListarEmpleados**
- + **dbo.P_AdminListarEntidades**
- + **dbo.P_AdminListarUsuarios**
- + **dbo.P_AdminRegistrarEmpleados**
- + **dbo.P_AdminRegistrarEntidad**
- + **dbo.P_AdminRegistrarUsuario**
- + **dbo.P_AdminValidarClave**

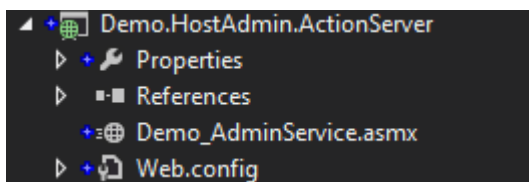
DOCUMENTO TECNICO DE APLICACIÓN PARA LA PRUEBA DE SELECCIÓN

BackEnd

El back end de la aplicación fue desarrollada en .Net con el lenguaje de C# utilizando capas en el modelo de arquitectura

- Hosting Layer
- Service Layer

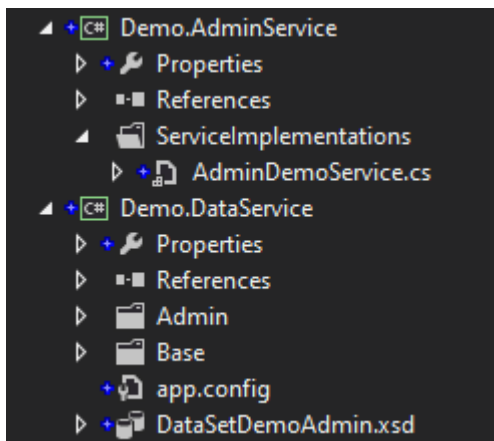
En el hosting layer encontramos el servicio que será publicado con su respectivo config de cadena de conexión



En el Service Layer encontramos la lógica del servicio y la extracción de los sp de la base de datos dividiendo esta capa en dos procesos

- AdminService
- DataService

El admin service nos almacena y ejecuta la lógica de los servicios, configurando los parámetros de entrada de cada método y los retornos se darán en formato Json para una serialización más sencilla



DOCUMENTO TECNICO DE APLICACIÓN PARA LA PRUEBA DE SELECCIÓN

Este es el primer método y se muestra su diseño de retorno para los demás métodos utilizados

```
[WebMethod]
[ScriptMethod(UseHttpGet = true, ResponseFormat = ResponseFormat.Json)]
public void P_AdminValidarClave(string Email, string Clave, string callback)
{
    ResultadoOperacion oResultadoOperacion = new ResultadoOperacion();

    oResultadoOperacion = _DataService.AdminValidarClave(Email, Clave);

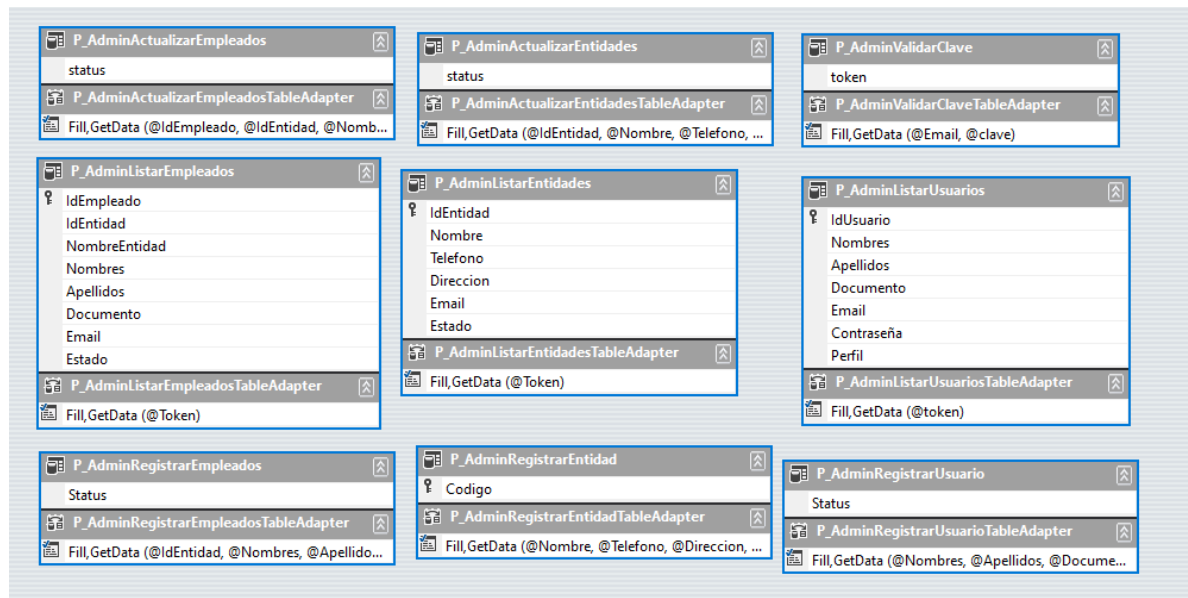
    if (oResultadoOperacion.oEstado == TipoRespuesta.Exito)
    {
        StringBuilder sb = new StringBuilder();
        JavaScriptSerializer js = new JavaScriptSerializer();
        js.MaxJsonLength = 2147483647;
        sb.Append(callback);
        sb.Append(js.Serialize(oResultadoOperacion));
        sb.Append("");

        Context.Response.Clear();
        Context.Response.ContentType = "application/json";
        Context.Response.Write(sb.ToString());
        Context.Response.End();
    }
    else
    {
        StringBuilder sb = new StringBuilder();
        JavaScriptSerializer js = new JavaScriptSerializer();
        sb.Append(callback);
        sb.Append(js.Serialize(oResultadoOperacion));
        sb.Append("");

        Context.Response.Clear();
        Context.Response.ContentType = "application/json";
        Context.Response.Write(sb.ToString());
        Context.Response.End();
    }
}
```

DOCUMENTO TECNICO DE APLICACIÓN PARA LA PRUEBA DE SELECCIÓN

En el data service encontramos la lógica de mapeo de entidades y la conexión de los procedimientos creados en la base de datos



```
public ResultadoOperacion AdminValidarClave(string Email, string Clave)
{
    ResultadoOperacion oResultadoOperacion = new ResultadoOperacion();
    string token = string.Empty;

    DataSetDemoAdmin.P_AdminValidarClaveDataTable _ValidarClave = new DataSetDemoAdmin.P_AdminValidarClaveDataTable();
    DataSetDemoAdmin.TableAdapters.P_AdminValidarClaveTableAdapter _ValidarClaveAdapter = new DataSetDemoAdmin.TableAdapters.P_AdminValidarClaveTableAdapter();

    try
    {
        _ValidarClave.Constraints.Clear();

        if (_ValidarClaveAdapter.Fill(_ValidarClave, Email, Clave) > 0)
        {
            oResultadoOperacion.oEstado = TipoRespuesta.Exito;
            oResultadoOperacion.Mensaje = "usuario OK";

            for (int i = 0; i < _ValidarClave.Rows.Count; i++)
            {
                token = _ValidarClave.Rows[i][0].ToString();

                oResultadoOperacion.EntidadDatos = token;
                oResultadoOperacion.oEstado = TipoRespuesta.Exito;
            }
        }
        else
        {
            oResultadoOperacion.oEstado = TipoRespuesta.Error;
            oResultadoOperacion.Mensaje = "Error al validar usuario";
        }
    }
    catch (Exception ex)
    {
        // Generar LOG DataBase Exception
        oResultadoOperacion.Mensaje = "Usuario o clave invalidos";
        oResultadoOperacion.oEstado = TipoRespuesta.Error;
    }

    return oResultadoOperacion;
}
```

DOCUMENTO TECNICO DE APLICACIÓN PARA LA PRUEBA DE SELECCIÓN

Cada uno de estos métodos utilizados genera su interfaz para conectarla al servicio y devolver todo por medio de mensajes que generan

- Código de respuesta
- Tipo de respuesta
- Información de la respuesta

DOCUMENTO TECNICO DE APLICACIÓN PARA LA PRUEBA DE SELECCIÓN

FrontEnd

Con el frontend encontramos toda la parte grafica y visual que el usuario podrá interactuar y validar, haciendo uso del backend ya publicado, esta parte se desarrolló en .net

Iniciar sesión

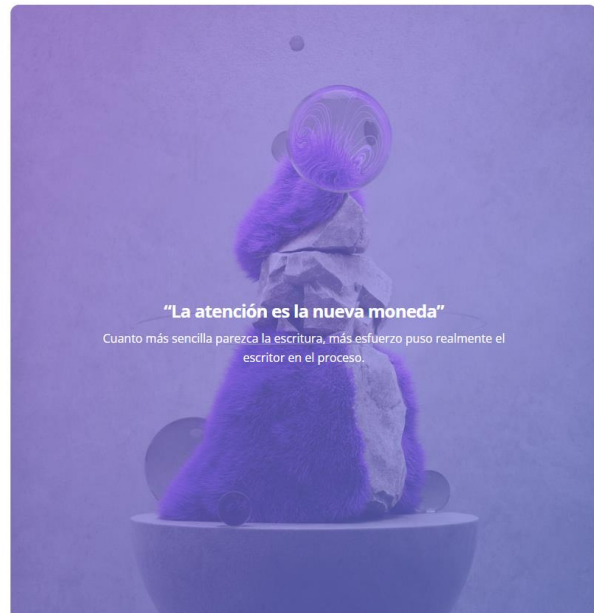
Ingresa tu email y contraseña para iniciar sesión

Contraseña

☐ Acuérdame de mí

Iniciar sesión

¿No tienes una cuenta? [Regístrate](#)



Dashboard

Empleados

Entidad

Usuarios

Pages / Dashboard

Dashboard

Type here...

Sign In

Lista de Empleados

Nuevo Empleado

NOMBRE COMPLETO	DOCUMENTO	EMAIL	ENTIDAD	ESTADO	
primero segundo	123456782	geotov@gmail.com	entidad1	Activo	Edite
Daniel Miranda	1033	miranded@gmail.com	entidad1	Activo	Edite

© 2024, made with by **Creative Tim** for a better web.

Creative TimAbout UsBlogLicense

DOCUMENTO TECNICO DE APLICACIÓN PARA LA PRUEBA DE SELECCIÓN

CI / CD

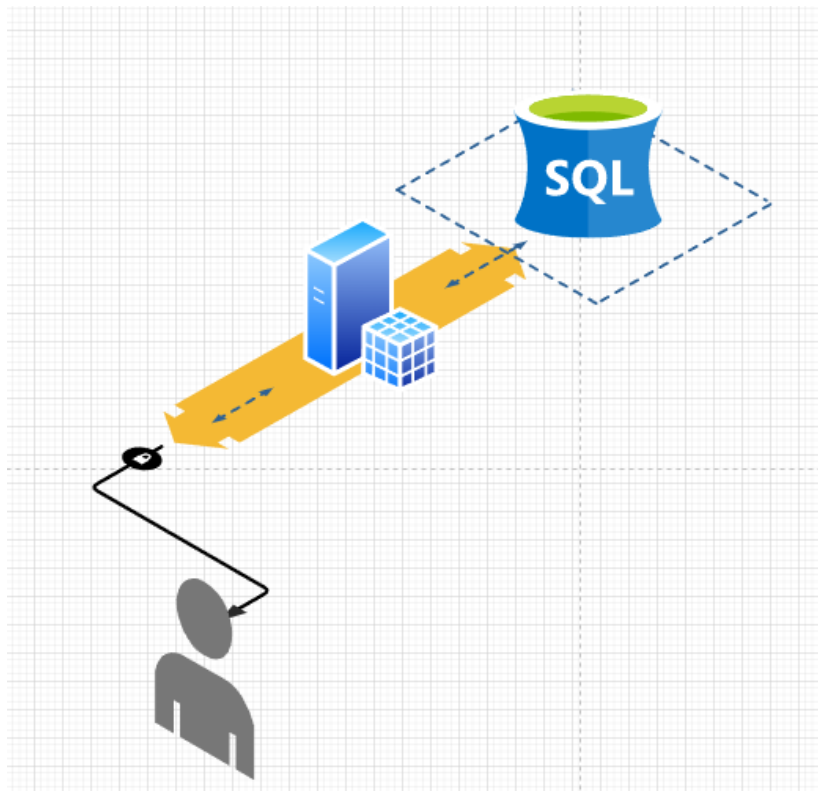
Con el fin de garantizar la integración continua y teniendo en cuenta el modelo de arquitectura realizado es posible garantizar cualquier tipo de integración a la plataforma sin afectar su origen base o inicial ya que permite agregar más módulos y así mismo generar más métodos al servicio que ejecuten nuevas funcionalidades garantizando los principios de esta integración que son

- Menos riesgos
- Mejor calidad
- Despliegues más rápidos

De igual forma garantizando el proceso de entrega continua al permitir actualizar por módulos la plataforma no se verán afectados los demás módulos y sus despliegues serán rápidos y limpios generando actualizaciones rápidas y con calidad conservando los principios de

- Automatización completa
- Rápida respuesta al cliente
- Feedback rápidos

Diagrama de arquitectura



DOCUMENTO TECNICO DE APLICACIÓN PARA LA PRUEBA DE SELECCIÓN

