


```

function closeModal(option) {
    $("button[data-dismiss='modal']").click();
    $('#modal-body').html("");
    modifyAlertsClasses(option);
}

function modifyAlertsClasses(option) {
    $('#successMessage').addClass('hidden');
    $('#editMessage').addClass('hidden');
    $('#deleteMessage').addClass('hidden');

    if (option === 'create') {
        $('#successMessage').removeClass('hidden');
    }
    else if (option === 'edit') {
        $('#editMessage').removeClass('hidden');
    }
    else if (option === 'delete') {
        $('#deleteMessage').removeClass('hidden');
    }
}

```

```

})(window.cibertec = window.cibertec || {});

```

4. Modifica el fichero "App\customer.js":

```

(function (customer) {
    customer.success = successReload;

    return customer;

    function successReload(option) {
        cibertec.closeModal(option);
    }

})(window.customer = window.customer || {});

```

5. Para poder hacer la llamada correspondiente a los mensajes creados tenemos que agregar un parámetro en las llamadas "Ajax.BeginForm".

- (1) En la vista parcial "_Create":

```

@using (Ajax.BeginForm("Create", new { Controller = "Customer" },
    new AjaxOptions
    {
        HttpMethod = "POST",
        InsertionMode = InsertionMode.Replace,
        UpdateTargetId = "modal-body",
        OnSuccess = "customer.success('create');",
    },
    htmlAttributes: new { id = "createForm" })))

```

- (2) En la vista parcial "_Edit":

```

@using (Ajax.BeginForm("Edit", new { Controller = "Customer" },
    new AjaxOptions
    {
        HttpMethod = "POST",
        InsertionMode = InsertionMode.Replace,
        UpdateTargetId = "modal-body",
        OnSuccess = "customer.success('edit');",
    },
    htmlAttributes: new { id = "editForm" })))

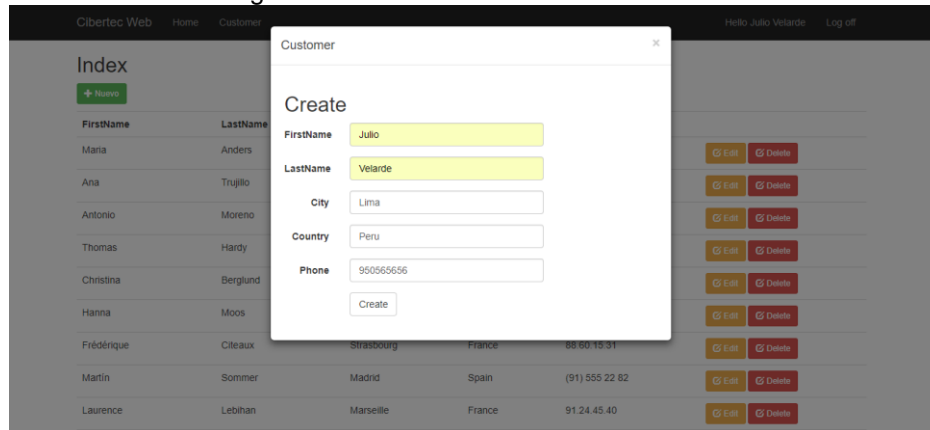
```

(3) En la vista parcial “_Delete”:

```
@using (Ajax.BeginForm("Delete", new { Controller = "Customer" },
    new AjaxOptions
    {
        HttpMethod = "POST",
        InsertionMode = InsertionMode.Replace,
        UpdateTargetId = "modal-body",
        OnSuccess = "customer.success('delete');",
    }, htmlAttributes: new { id = "deleteForm" })))
```

6. Procedemos a ejecutar nuestra aplicación y validamos los resultados:

(1) Al crear un nuevo registro:

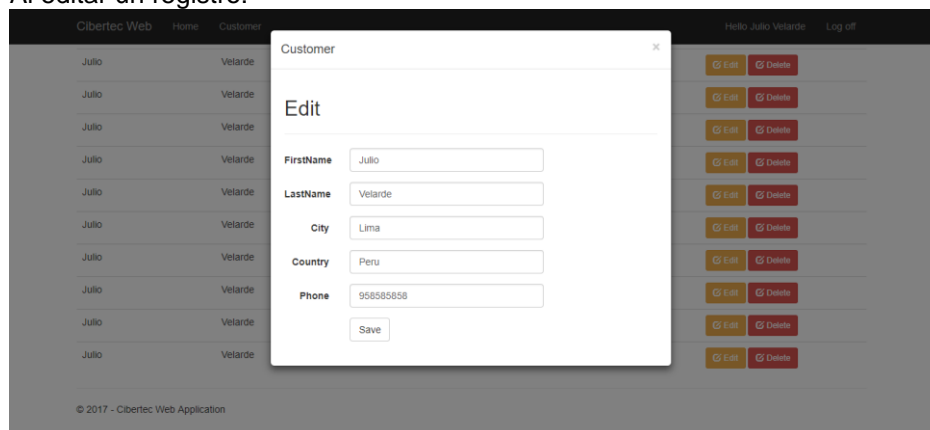


Resultado esperado:

Index



(2) Al editar un registro:

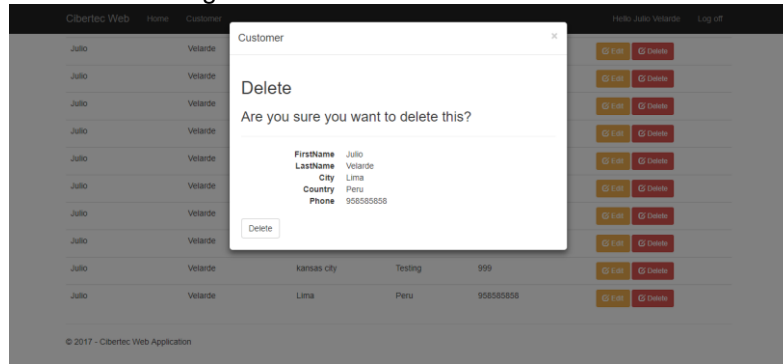


Resultado esperado:

Index



(3) Al eliminar un registro:



Resultado esperado:

Index



IV. EVALUACIÓN

1. ¿Cuáles son las carpetas principales en una aplicación MVC?

Las carpetas principales son: Controllers, Views y Models

2. ¿Qué pasa si al nombre del controlador no se le coloca el prefijo Controller?

Por convención todos los archivos que van a ser controladores deben tener el prefijo Controller al final del nombre para que puedan ser reconocidos por Asp.NET como tales.

3. ¿Cuál es el tipo de datos de los métodos del controller?

El tipo base de todos los controladores es ActionResult.

4. ¿Qué es una vista Layout?

Es la vista maestra, en ella se establece la estructura base (html) de todas las vistas que tendrá la aplicación web. Algunas vistas pueden no utilizar la vista base.

5. Los modelos, ¿pueden ser librería de clases?

Sí, los modelos son un concepto que involucra entidades, acceso a datos, reglas de negocio y pueden ser implementadas en librería de clases (dll).

6. ¿Qué son las vistas parciales?

Al igual que las vistas normales, estas permiten ser reutilizadas en ciertas partes de la aplicación.