Construir una página web que cumpla los siguientes criterios

* Contenga un formulario (\Views\Personas\Create.cshtml), que estos campos están validados de acuerdo al tipo de dato que es (ejemplo: Nombre, solo letras) (\Models\Persona.cs líneas 11-15)
* Que este formulario se visualice correctamente en modo escritorio *y en móvil (\Views\Shared\\_Layout.cshtml línea 35)*
* El botón guardar solo se habilitará si están llenos todos los campos (\Views\Personas\Create.cshtml líneas 57-63)
* Que al dar click en el botón guardar se deben de enviar los datos a un API el cual valide los datos enviados(\Views\Personas\Create.cshtml líneas 110-136 7-55),

\* Si son correctos los datos, guardarlos en una base de datos (“Datos guardados correctamente”) Mostrar un modal con los datos que se guardaron y al dar en aceptar: Muestra una tabla con los registros realizados.

\* De ser incorrectos los datos, mostrar un modal con el dato que no pasó las validaciones remarcado en rojo (Error).

Infraestructura de construcción proyecto

● Deben estar construidos en **REST API (C#)** (\Controllers\PersonasApiController.cs)

● Contener una web app mediante el cual se podrá acceder al proyecto y este debe ser desarrollado usando HTML5, CSS3 (\wwwroot\css\site.css), JavaScript, JS(JQUERY) (\Views\Personas\Create.cshtml líneas 3-71) **Nota**: Se puede hacer uso de framework (el proyecto ya usaba Bootstrap)

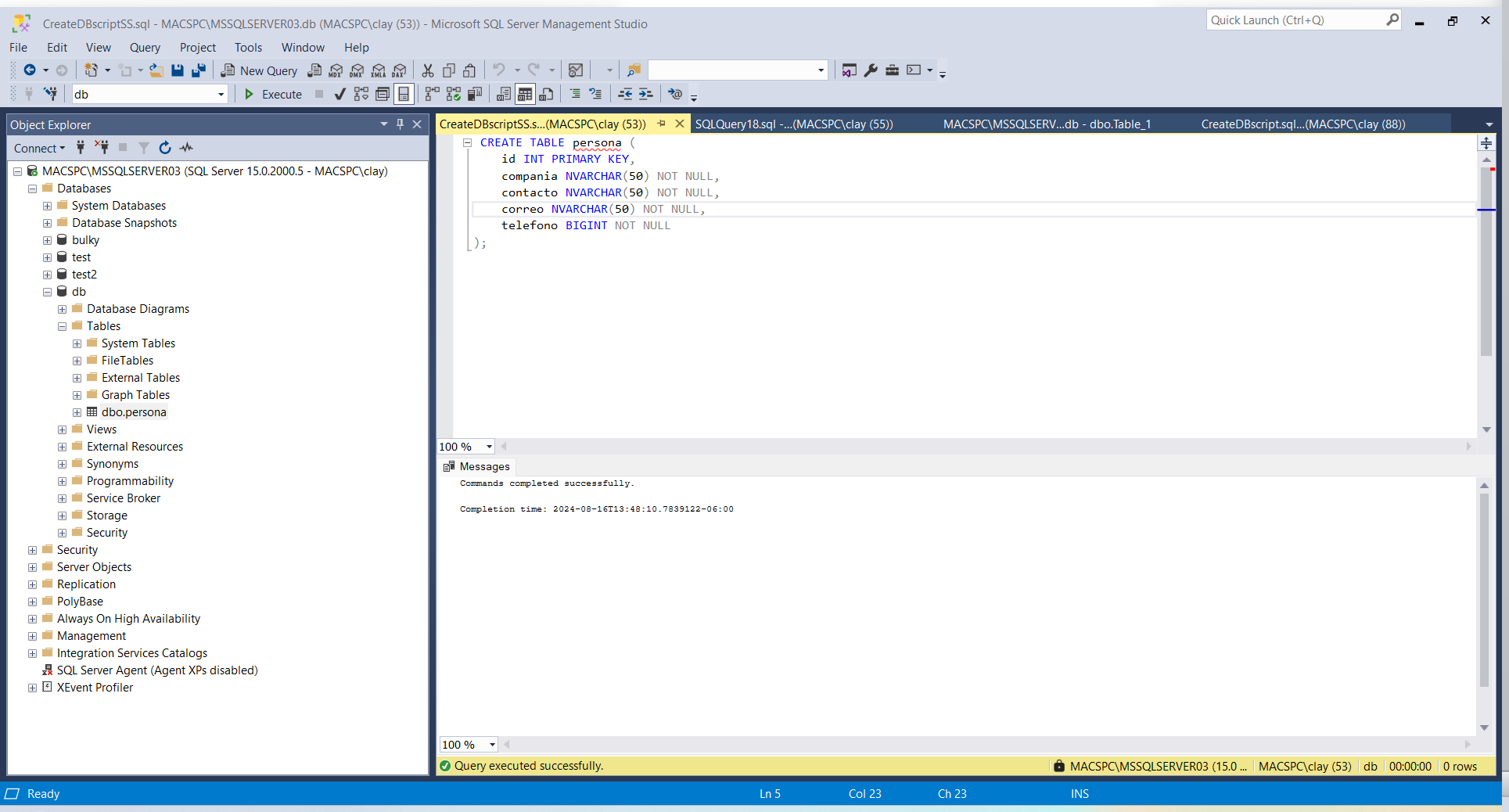
● El proyecto se puede compartir en in Git o carpeta comprimida. (https://github.com/macs04/Claims)

● Objetos de base de datos compartir script para su implementación (/CreateDBscriptSS.sql)

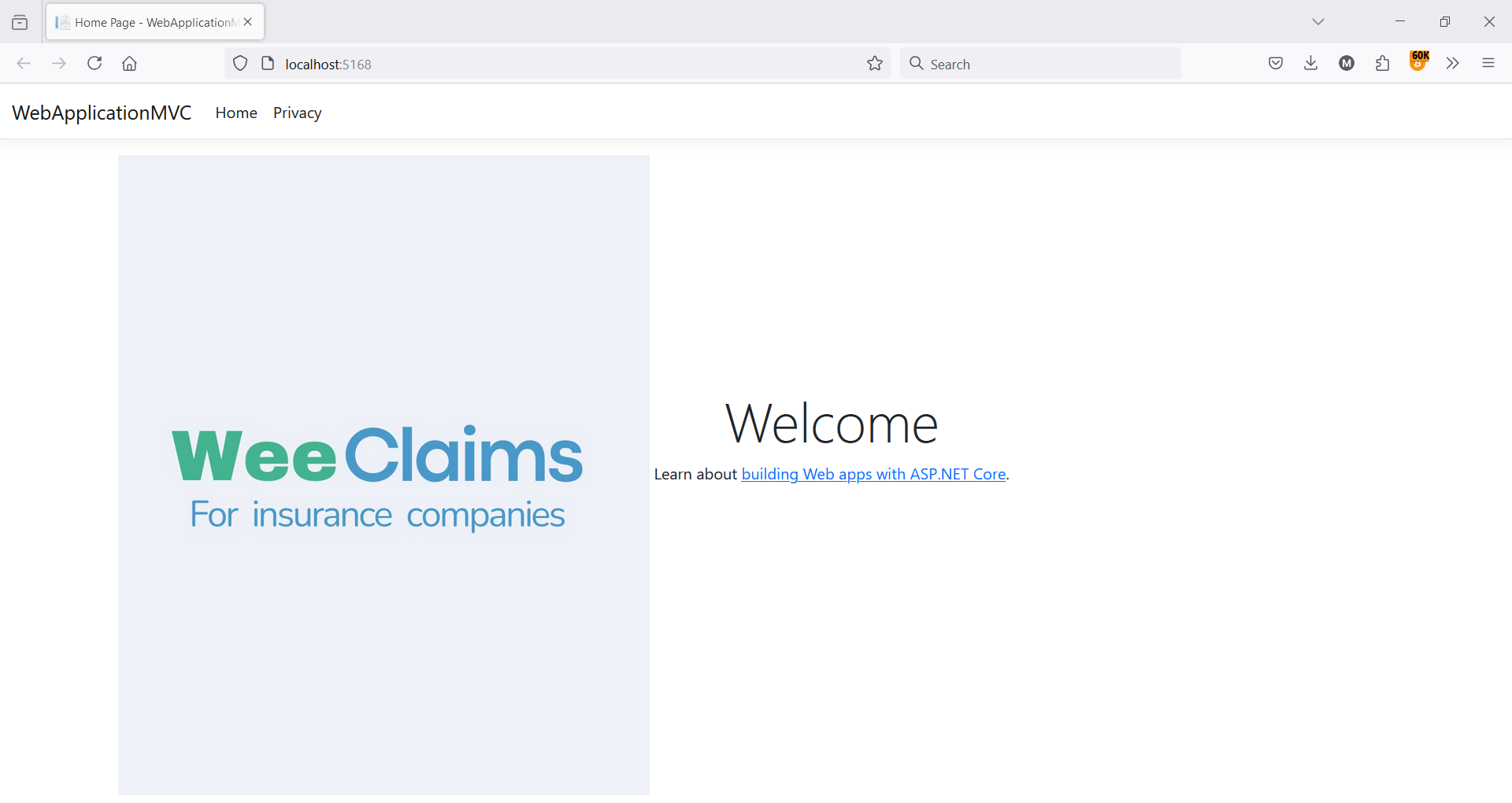
● Agregar las instrucciones de ejecución

Instrucciones

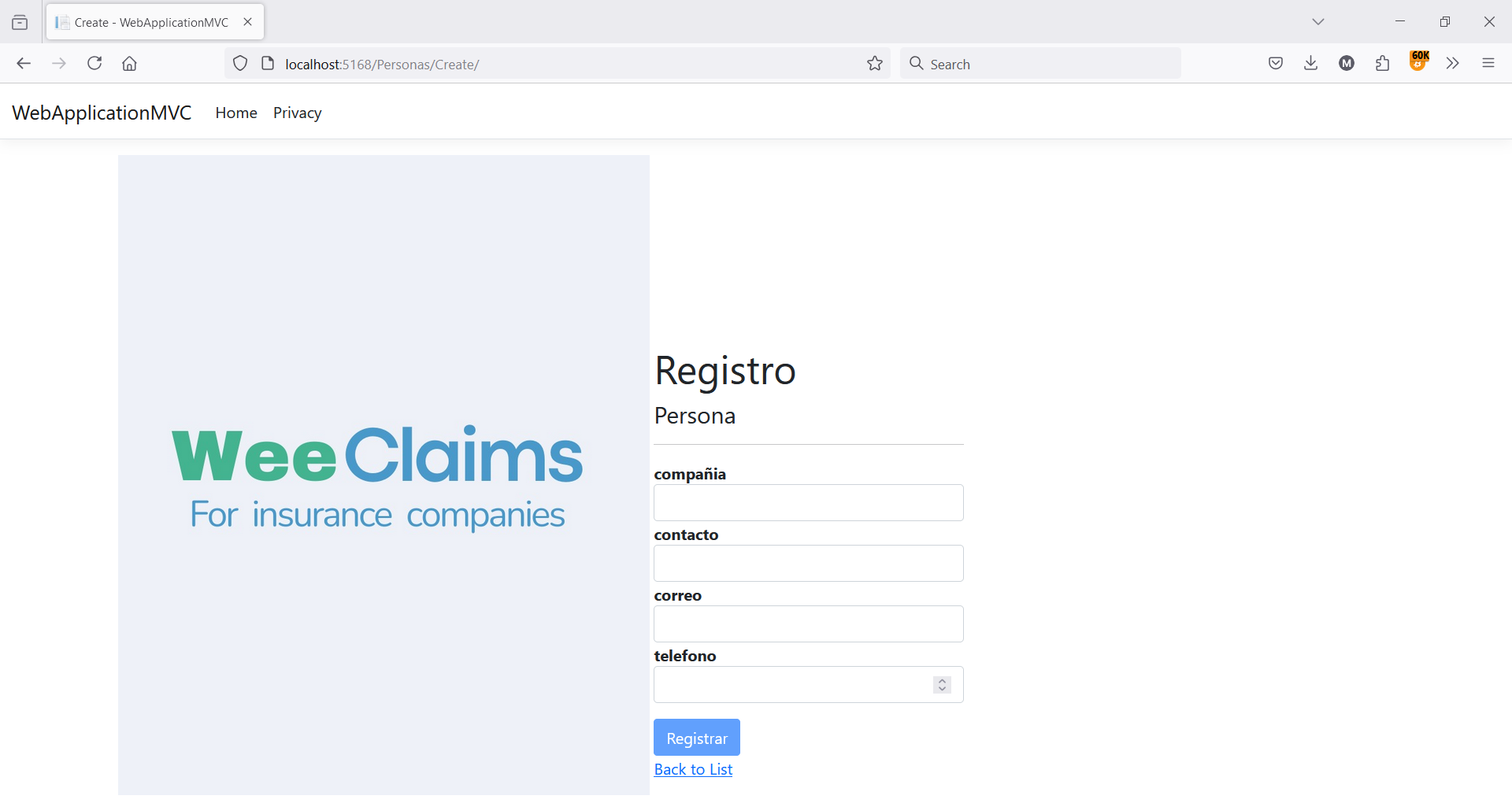
En SQL Server, abrir, ejecutar /CreateDBscriptSS.sql y recargar para ver la tabla.



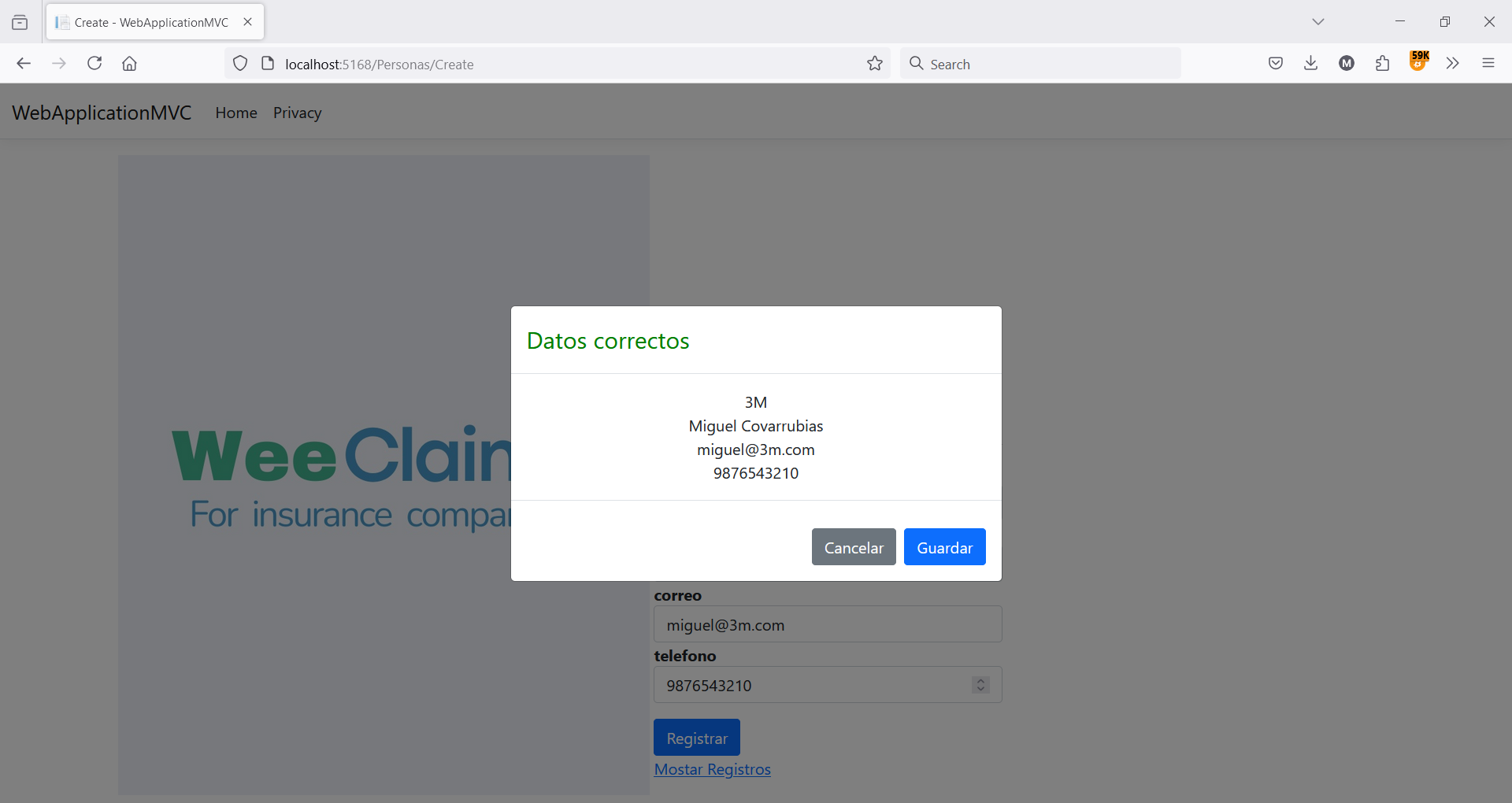
Ejecutar el proyecto / WebApplicationMVC con F5



Navegar a <http://localhost:5168/Personas/Create>

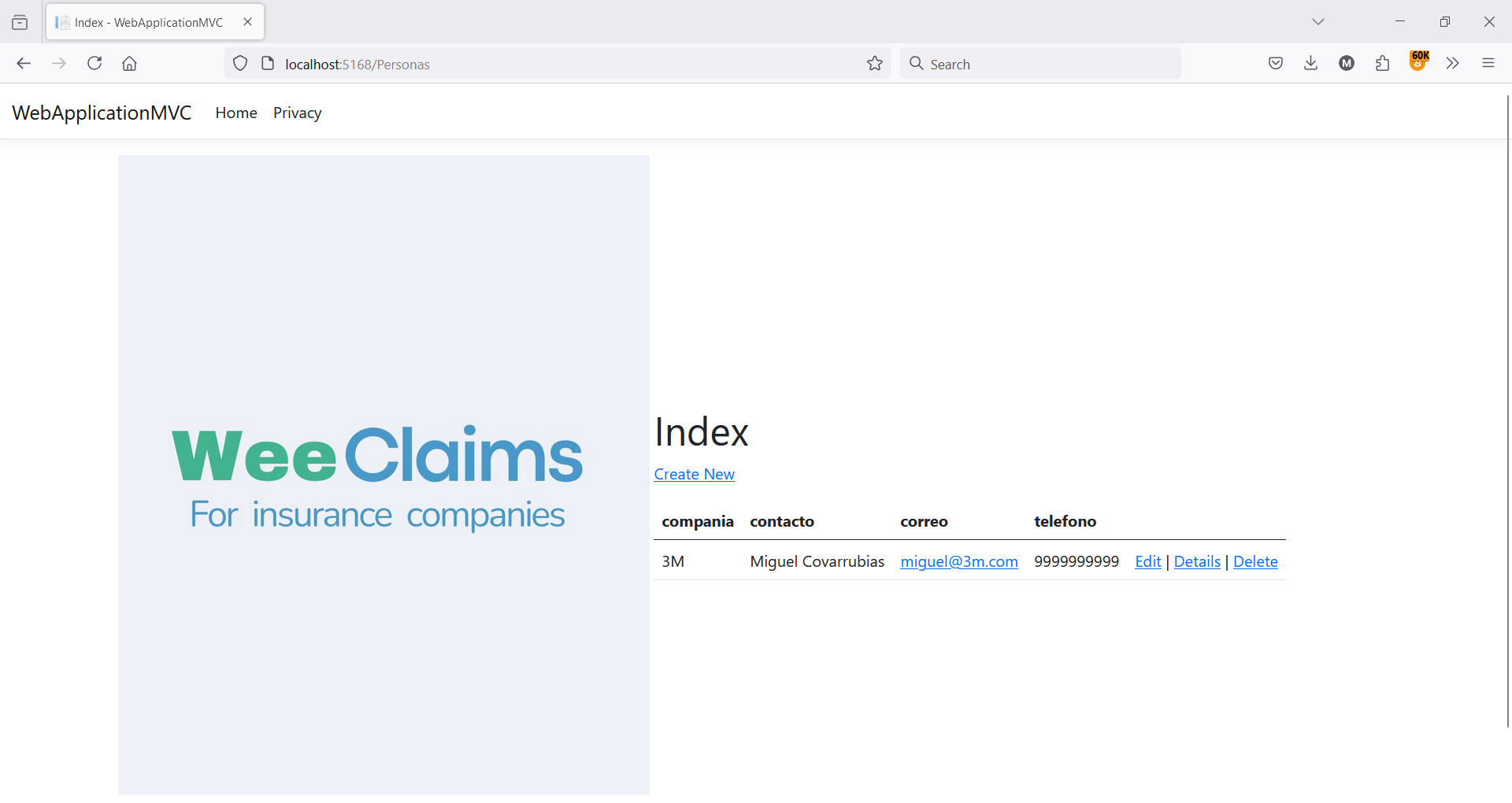


Registrar a una persona con Registrar->Guardar para guardar.

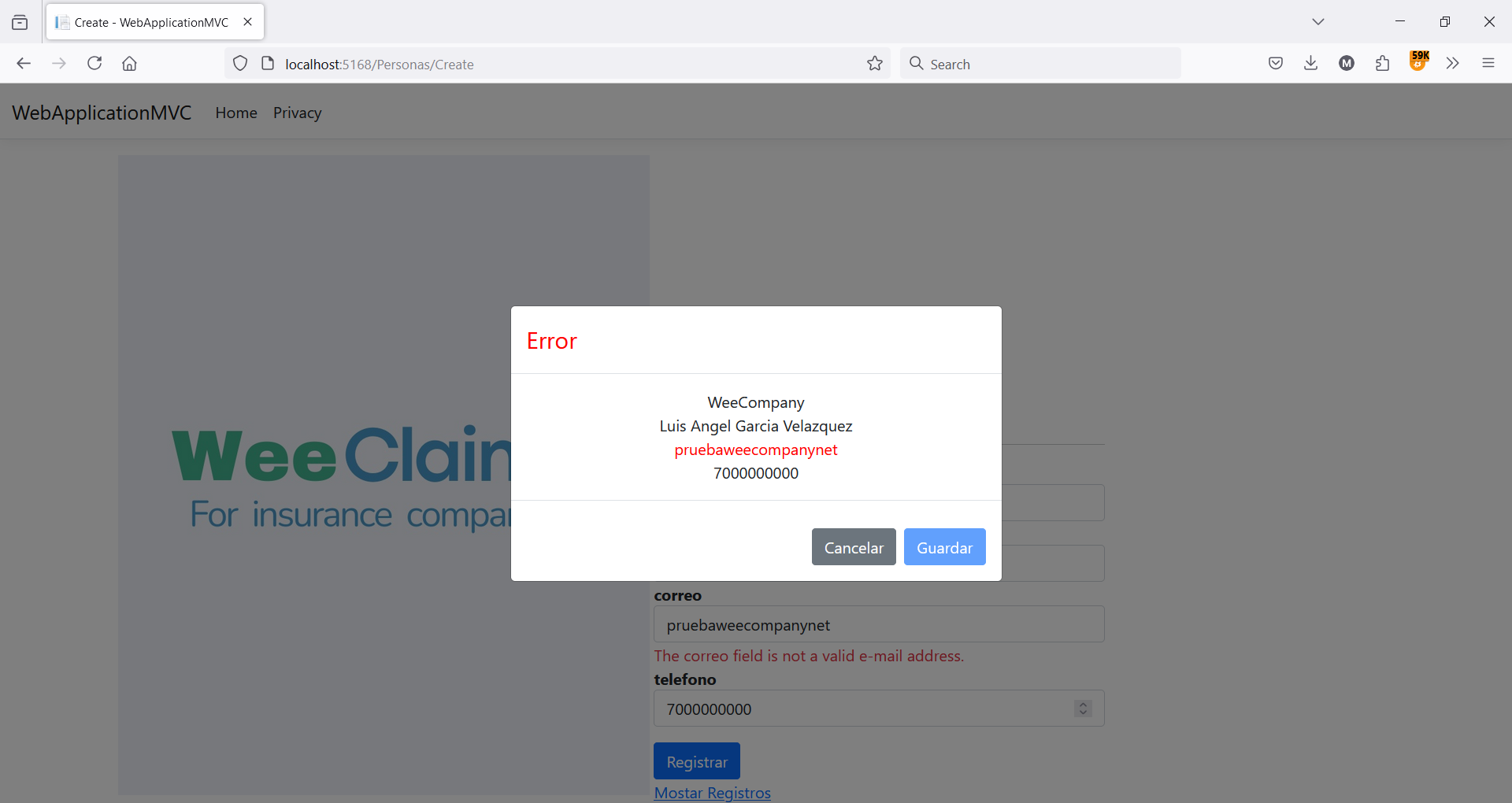


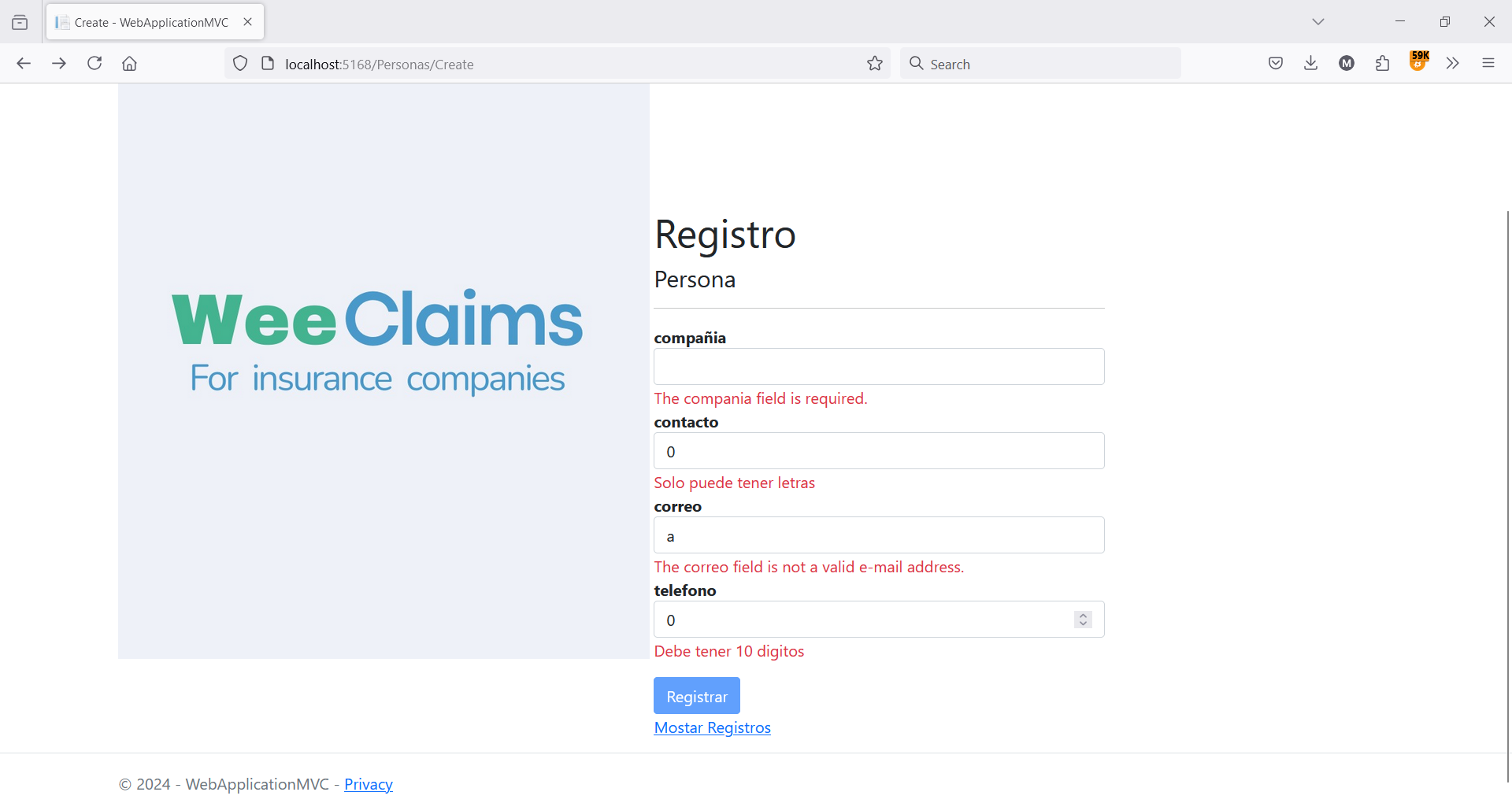
Modal

Verificar que se guardo en la base de datos haciendo click en Mostrar Registros, ahí también se puede editar, ver detalles o borrar cualquier registro.



Tratar de crear un nuevo registro que no sea valido

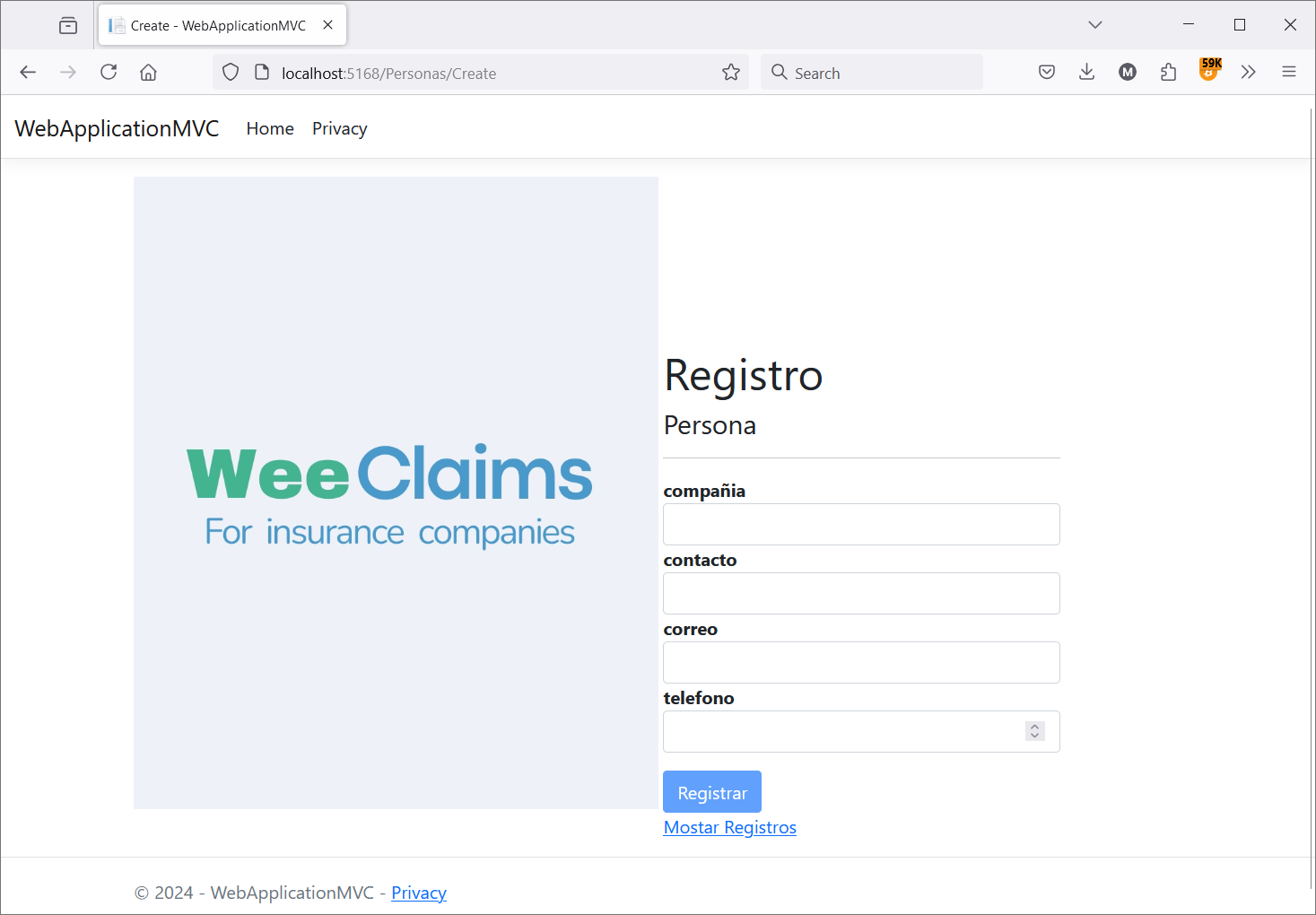
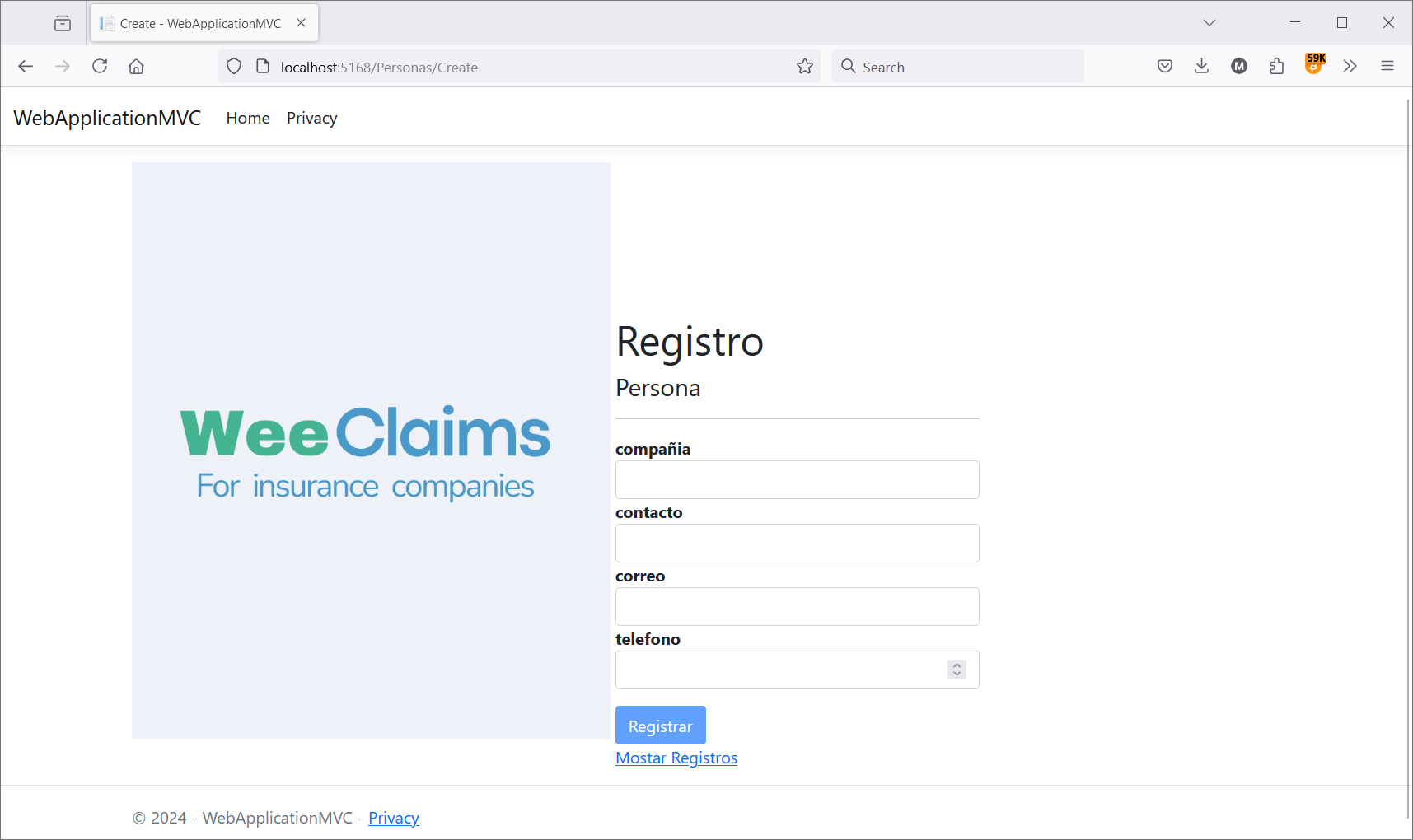




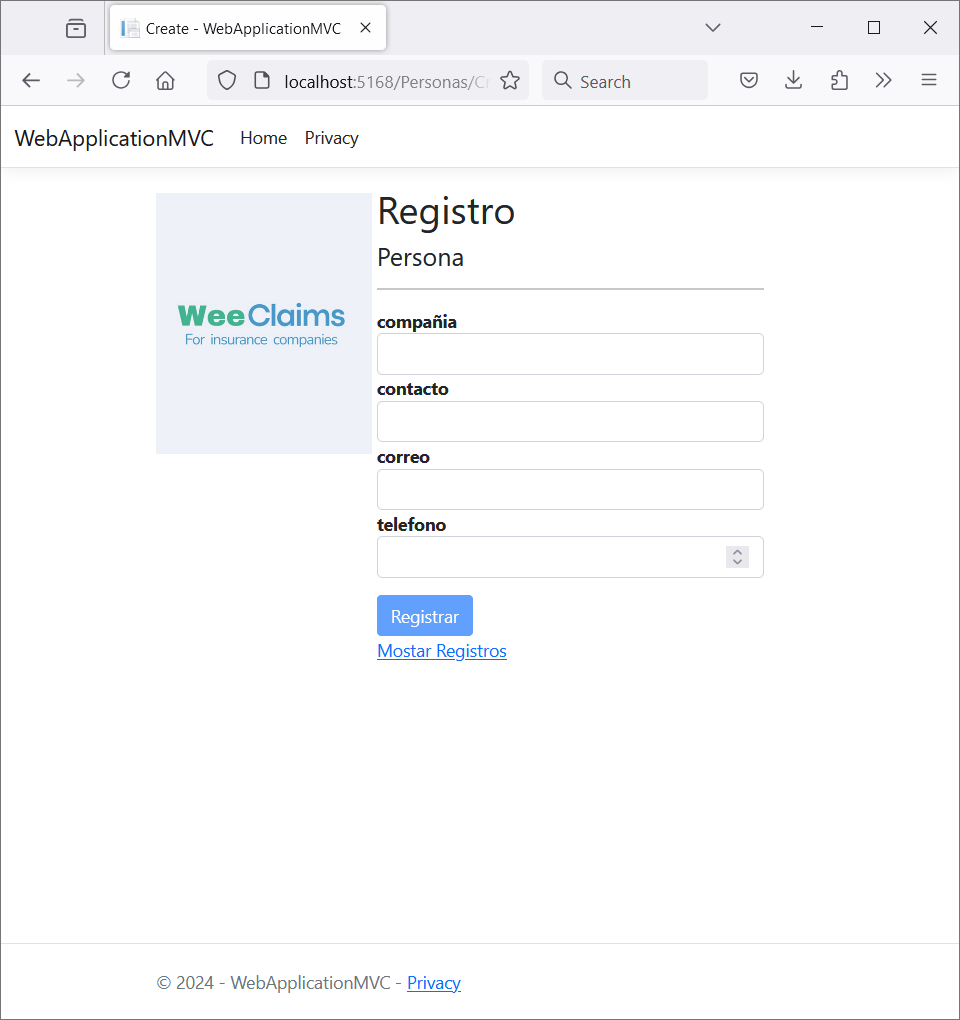
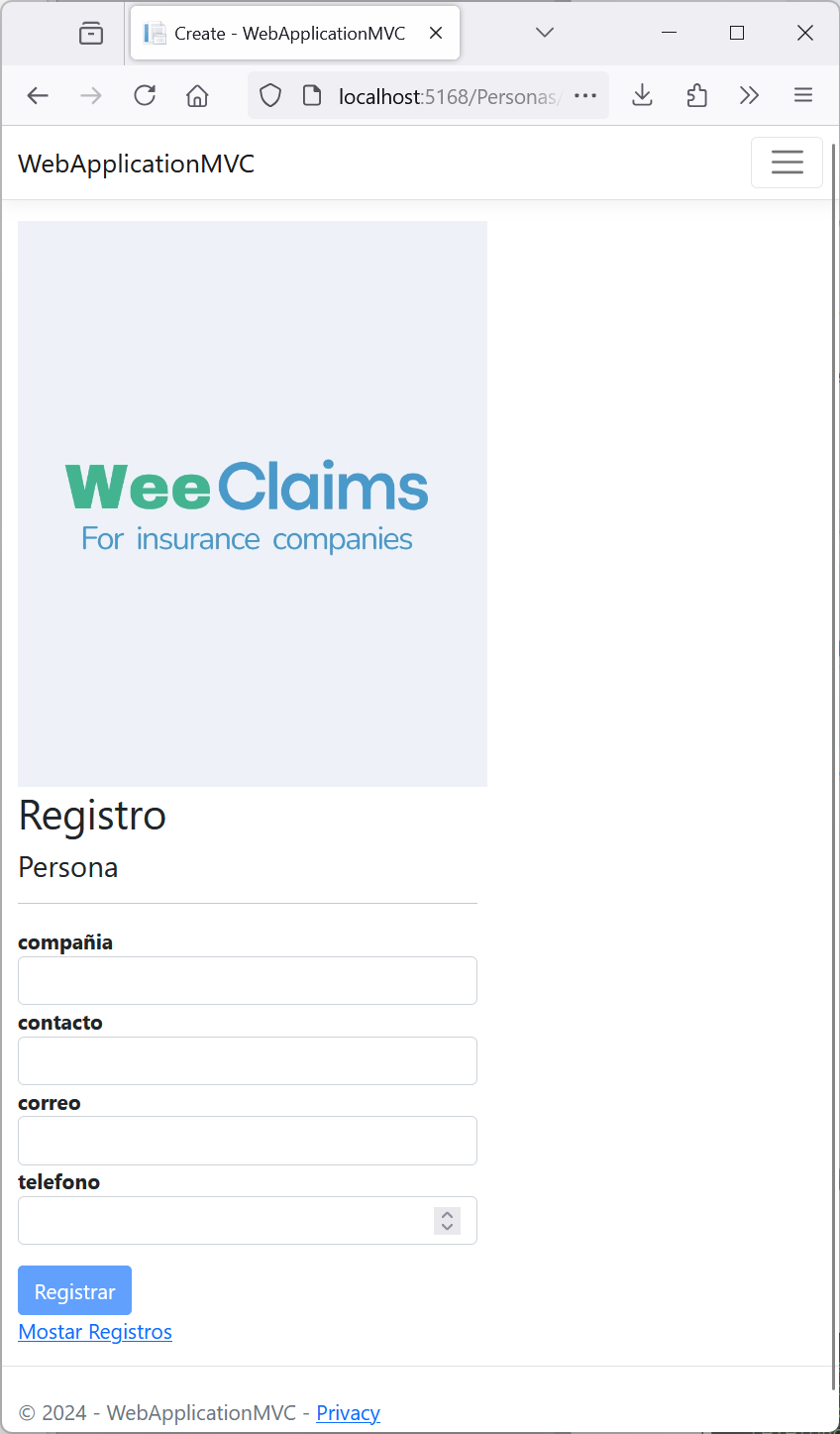
Validaciones

Para la API, navegar a <http://localhost:5168/swagger>. Desde el swagger, se pueden registrar personas, obtener todas o alguna persona en especifico, actualizar personas y borrar personas. En el recuadro muestro una ejecución de cada endpoint.

Vistas



Large Medium, Laptop

Small Extra small, Celular

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| POST  /api/PersonasApi  **Parameters**  No parameters  **Request body**   {  "id": 0,  "compania": "string",  "contacto": "hbeWvovUepvKQMDUmvdrBfUEsTNbqPA",  "correo": "user@example.com",  "telefono": 9999999999  }  **Responses**  **Curl**  curl -X 'POST' \  'http://localhost:5168/api/PersonasApi' \  -H 'accept: \*/\*' \  -H 'Content-Type: application/json' \  -d '{  "id": 0,  "compania": "string",  "contacto": "hbeWvovUepvKQMDUmvdrBfUEsTNbqPA",  "correo": "user@example.com",  "telefono": 9999999999  }'  **Request URL**  http://localhost:5168/api/PersonasApi  **Server response**   | Code | Details | | --- | --- | | 200 | **Response body** |  |  | | --- | | {  "id": 0,  "compania": "string",  "contacto": "hbeWvovUepvKQMDUmvdrBfUEsTNbqPA",  "correo": "user@example.com",  "telefono": 9999999999  }  **Response headers**  content-type: application/json; charset=utf-8 date: Fri,16 Aug 2024 21:29:09 GMT server: Kestrel transfer-encoding: chunked |   **Responses**   | Code | Description | Links | | --- | --- | --- | | 200 | OK | *No links* |   GET  /api/PersonasApi  **Parameters**  No parameters  **Responses**  **Curl**  curl -X 'GET' \  'http://localhost:5168/api/PersonasApi' \  -H 'accept: \*/\*'  **Request URL**  http://localhost:5168/api/PersonasApi  **Server response**   | Code | Details | | --- | --- | | 200 | **Response body** |  |  | | --- | | [  {  "id": 0,  "compania": "string",  "contacto": "hbeWvovUepvKQMDUmvdrBfUEsTNbqPA",  "correo": "user@example.com",  "telefono": 9999999999  }  ]  **Response headers**  content-type: application/json; charset=utf-8 date: Fri,16 Aug 2024 21:29:17 GMT server: Kestrel transfer-encoding: chunked |   **Responses**   | Code | Description | Links | | --- | --- | --- | | 200 | OK | *No links* |   GET  /api/PersonasApi/{id}  **Parameters**   | **Name** | **Description** | | --- | --- | | id \*  integer($int32)  (path) |  |   **Responses**  **Curl**  curl -X 'GET' \  'http://localhost:5168/api/PersonasApi/0' \  -H 'accept: \*/\*'  **Request URL**  http://localhost:5168/api/PersonasApi/0  **Server response**   | Code | Details | | --- | --- | | 200 | **Response body** |  |  | | --- | | {  "id": 0,  "compania": "string",  "contacto": "hbeWvovUepvKQMDUmvdrBfUEsTNbqPA",  "correo": "user@example.com",  "telefono": 9999999999  }  **Response headers**  content-type: application/json; charset=utf-8 date: Fri,16 Aug 2024 21:29:28 GMT server: Kestrel transfer-encoding: chunked |   **Responses**   | Code | Description | Links | | --- | --- | --- | | 200 | OK | *No links* |   PUT  /api/PersonasApi/{id}  **Parameters**   | **Name** | **Description** | | --- | --- | | id \*  integer($int32)  (path) |  |   **Request body**   {  "compania": "nueva compañia",  "contacto": "HeOqbEItKlVZbfdwMXpzQTlgPCzKYRtkXCj JlZMImrtZXCjUkeuRHeucyKNfQkEIbLxNrHDrMwmImhIOQFPseCHCvppxe",  "correo": "user@example.com",  "telefono": 9999999999  }  **Responses**  **Curl**  curl -X 'PUT' \  'http://localhost:5168/api/PersonasApi/0' \  -H 'accept: \*/\*' \  -H 'Content-Type: application/json' \  -d '{  "compania": "nueva compañia",  "contacto": "HeOqbEItKlVZbfdwMXpzQTlgPCzKYRtkXCj JlZMImrtZXCjUkeuRHeucyKNfQkEIbLxNrHDrMwmImhIOQFPseCHCvppxe",  "correo": "user@example.com",  "telefono": 9999999999  }'  **Request URL**  http://localhost:5168/api/PersonasApi/0  **Server response**   | Code | Details | | --- | --- | | 200 | **Response body** |  |  | | --- | | {  "id": 0,  "compania": "nueva compañia",  "contacto": "HeOqbEItKlVZbfdwMXpzQTlgPCzKYRtkXCj JlZMImrtZXCjUkeuRHeucyKNfQkEIbLxNrHDrMwmImhIOQFPseCHCvppxe",  "correo": "user@example.com",  "telefono": 9999999999  }  **Response headers**  content-type: application/json; charset=utf-8 date: Fri,16 Aug 2024 21:30:28 GMT server: Kestrel transfer-encoding: chunked |   **Responses**   | Code | Description | Links | | --- | --- | --- | | 200 | OK | *No links* |   DELETE  /api/PersonasApi/{id}  **Parameters**   | **Name** | **Description** | | --- | --- | | id \*  integer($int32)  (path) |  |   **Responses**  **Curl**  curl -X 'DELETE' \  'http://localhost:5168/api/PersonasApi/0' \  -H 'accept: \*/\*'  **Request URL**  http://localhost:5168/api/PersonasApi/0  **Server response**   | Code | Details | | --- | --- | | 200 | **Response headers**  content-length: 0 date: Fri,16 Aug 2024 21:30:55 GMT server: Kestrel |   **Responses**   | Code | Description | Links | | --- | --- | --- | | 200 | OK |  | |