



## **MANUAL TÉCNICO**

# **“SISTEMA INFORMÁTICO DE RECEPCIÓN DE ATESTADOS”**

## **ELABORADO POR**

Andrés Cornejo Hidalgo  
Andrés Irías Li  
Carlos Kruse Alvarenga  
Francisco Badilla Fonseca

**SAN JOSÉ  
2022**

# RESUMEN

En este documento se detalla toda la información técnica referente al proyecto SIRA, o Sistema Informático de Recepción de Atestados. Este sistema es una aplicación web diseñada con el propósito de facilitar a los funcionarios del ITCR y a los miembros de la Comisión de Carrera Profesional, la entrega, registro y evaluación de los atestados que dichos funcionarios deseen someter a evaluación. Esta aplicación se inició el año 2021 y este proyecto consiste en darle mantenimiento y expandir sus funcionalidades de forma que cumpla los requerimientos para los cuales fue creado originalmente. Para desarrollar este proyecto se utilizó la metodología SCRUM con tres Sprints de cuatro semanas cada uno.

**Palabras clave:** SIRA, Comisión de Carrera Profesional, ITCR, SQL Server, .NET Framework

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Sobre este manual . . . . .	1
1.2. Descripción y Alcance del proyecto . . . . .	1
1.2.1. Antecedentes . . . . .	1
1.2.2. Objetivos . . . . .	1
1.2.3. Contacto . . . . .	2
<b>2. Especificación de funcionalidad</b>	<b>3</b>
2.1. Descripción del producto . . . . .	3
2.1.1. Historias de usuario . . . . .	3
<b>3. Arquitectura del sistema</b>	<b>7</b>
3.1. Diseño general del sistema . . . . .	7
3.1.1. Diseño de la persistencia . . . . .	7
3.1.2. Diagrama de clases . . . . .	9
<b>4. Instrucciones de Instalación</b>	<b>10</b>
4.1. Entorno de Desarrollo y Producción . . . . .	10
4.1.1. Dependencias . . . . .	10
4.1.2. Instrucciones de instalación . . . . .	11
4.1.3. Acceso al Servidor . . . . .	12
<b>5. Métodos del API</b>	<b>14</b>
<b>6. Conclusiones y trabajo futuro</b>	<b>16</b>
6.1. Conclusiones . . . . .	16
6.2. Problemáticas y limitaciones . . . . .	16
6.3. Trabajo futuro . . . . .	16
<b>Referencias bibliográficas</b>	<b>17</b>

# Índice de figuras

3.1. Diagrama entidad-relación de la base de datos del sistema. . . . .	7
3.2. Diagrama relacional del sistema. . . . .	8
3.3. Diagrama de clases del sistema. . . . .	9

# Capítulo 1

## Introducción

### 1.1. Sobre este manual

El presente documento describe los aspectos técnicos del Sistema Informático de Recepción de Atestados, SIRA de ahora en adelante, desarrollado anteriormente por los estudiantes Andrés Castro y Gerarld Sánchez y posteriormente por Francisco Badilla, Andrés Cornejo, Andrés Irías y Carlos Kruse del Instituto Tecnológico de Costa Rica. El objetivo de este documento es servir como una introducción al sistema de nuevas personas que vayan a realizar algún trabajo en el aspecto técnico, como por ejemplo programador, arquitecto de software o personal de TI que necesita administrar la aplicación.

### 1.2. Descripción y Alcance del proyecto

#### 1.2.1. Antecedentes

El **Sistema Informático de Recepción de Atestados** es un proyecto que se ha estado desarrollando gracias a una colaboración entre la **Comisión de Carrera Profesional** y los estudiantes que forman parte del curso de Proyecto de Ingeniería de Software. Dicho sistema se encarga de registrar documentos catalogados como atestados que demuestran el desempeño profesional de los diferentes funcionarios de la institución. Si bien, esta funcionalidad ya ha sido implementada en semestres anteriores, surge la necesidad de terminar de completar las funcionalidades planteadas para este sistema. De manera paralela, existe la necesidad de implementar un módulo complementario que se encargue de facilitar el proceso de revisión de los atestados.

#### 1.2.2. Objetivos

- **Objetivo General**

- Desarrollar un sistema que facilite y automatice los procesos de envío, recepción y evaluación de atestados para la Comisión en un transcurso de 12 semanas.

- **Objetivos Específicos**

- Programar desde el nivel más bajo posible, una conexión estándar para que los módulos desarrollados en el futuro puedan comunicarse entre ellos.
- Extender el módulo de presentación de atestados para los funcionarios del ITCR.
- Depurar la funcionalidad del módulo heredado de presentación de atestados para los funcionarios del ITCR.

- Crear un módulo que le permita a los funcionarios revisores de la comisión revisar y calificar todos los atestados presentados por los funcionarios del ITCR.
- Diseñar la interfaz gráfica y el código de los módulos bajo los estándares de diseño del DATIC.

### **1.2.3. Contacto**

Para consultas y soporte, contacte a: [andrewcornejoh@estudiantec.cr](mailto:andrewcornejoh@estudiantec.cr), [andresirias02@gmail.com](mailto:andresirias02@gmail.com), [carlos.roberto.kruse@gmail.com](mailto:carlos.roberto.kruse@gmail.com) o [francisco.badilla@estudiantec.cr](mailto:francisco.badilla@estudiantec.cr)

## Capítulo 2

# Especificación de funcionalidad

### 2.1. Descripción del producto

Para promover el desarrollo y el ingreso de personal profesional en el Instituto Tecnológico de Costa Rica (ITCR), se estableció un reglamento mediante el cual se reconocen los méritos académicos del personal y se retribuye de manera correspondiente. El **Reglamento de Carrera Profesional** plantea un sistema de categorías que permite evaluar los logros profesionales y académicos de cada funcionario y asignarles la categoría respectiva.

Este reglamento toma en cuenta obras que aporten valor académico, profesional o artístico según la disciplina a la que pertenece el funcionario, tales como libros, artículos, software, u otras producciones; al igual que el desempeño de actividades que demuestren iniciativa, como asistencia a capacitaciones profesionales y participación en diferentes departamentos de la institución.

Cada uno de estos méritos se debe someter a la evaluación de la **Comisión de Carrera Profesional** mediante atestados que demuestren su validez. Cada miembro de dicha comisión evalúa este atestado y le asigna una puntuación, la cual se promedia entre todos los miembros. Al acumular cierta cantidad de puntos, un funcionario se considera cualificado para ser admitido en la siguiente categoría.

#### 2.1.1. Historias de usuario

Pila de Funcionalidades del Sistema

Código	Descripción	Criterios de Calidad
RQ001	<b>Evaluar atestado:</b> Los revisores podrán acceder a los atestados de los docentes y emitir una evaluación correspondiente. Podrán ver los datos relevantes como el archivo adjunto, la fecha de envío, y la información del docente.	El usuario revisor puede observar correctamente el resultado de la evaluación emitida y editarlo.
RQ002	<b>Evaluación de Desempeño:</b> Los revisores podrán acceder a los atestados de los docentes y emitir una evaluación correspondiente. Podrán ver los datos relevantes como el archivo adjunto, la fecha de envío, y la información del docente.	El usuario revisor puede observar correctamente el resultado de la evaluación emitida y editarlo.

Código	Descripción	Criterios de Calidad
<b>RQ003</b>	<b>Ver Docente:</b> Los usuarios podrán acceder a un perfil que corresponde con un docente. Los revisores podrán ver los perfiles de los demás docentes, mientras que los funcionarios docentes sólo pueden acceder a su propio perfil. Ahí habrá información como los atestados presentados y el tiempo servido.	Los usuarios pueden averiguar la información de un docente de manera satisfactoria.
<b>RQ004</b>	<b>Barra de Progreso:</b> Al acceder al perfil de un docente, se podrá visualizar el puntaje del mismo por medio de una barra que mide el porcentaje de puntos que requiere para ser asignado a la siguiente categoría.	El usuario puede observar correctamente que la barra de progreso se actualice conforme se va evaluando.
<b>RQ005</b>	<b>Inicio de Sesión:</b> Funcionalidad que involucra el manejo de los datos ingresados por el usuario para verificar si se encuentra un usuario registrado en el sistema y su correspondiente verificación.	El usuario logra acceder satisfactoriamente a la aplicación si ingresa los credenciales correctos.
<b>RQ006</b>	<b>Rediseño de interfaz:</b> La interfaz gráfica original de la aplicación es suficiente, sin embargo, es muy monótona y a veces ambigua. Una nueva interfaz, que sea más clara, elegante y dinámica es algo que la aplicación necesita.	El usuario logra interactuar con la interfaz de manera satisfactoria y placentera.
<b>RQ007</b>	<b>CRUD de usuarios y personas:</b> La aplicación permite crear, editar y eliminar personas en las categorías de funcionario, administrador y revisor	La creación, modificación y borrado de un usuario ocurre de manera correcta sin errores.
<b>RQ008</b>	<b>Verificación por lista:</b> La aplicación verifica los documentos por medio de una lista, considerando datos como el nombre o su consecutivo correspondiente	El usuario logra encontrar satisfactorialmente el documento que esté buscando en la lista.
<b>RQ009</b>	<b>Trasladar atestado:</b> La aplicación permite al usuario cambiar o trasladar un atestado registrado de rubro	El usuario logra trasladar un atestado conservando la información esencial del atestado.
<b>RQ010</b>	<b>Asignación automática de puntos:</b> El sistema recibe datos para generar una asignación automática de puntos, esto según el tipo de atestado y el tipo de datos que se esté ingresando (autores, años, correos...)	La aplicación asigna los puntos de manera correcta. El usuario logra verificar que la aplicación logró asignar los puntos correspondientes sin errores en el cálculo.



Código	Descripción	Criterios de Calidad
<b>RQ011</b>	<b>Evaluar varios autores:</b> En los atestados donde sea posible, los revisores pueden asignar un puntaje a cada uno de los funcionarios que lo realizaron. Este puntaje puede ser repartido de cualquier forma.	Se puede asignar correctamente el porcentaje que recibe cada autor. un autor no puede obtener una cantidad mayor a la máxima permitida para un autor, si es que se tienen más puntos por ser varios autores.
<b>RQ012</b>	<b>Evaluación por medio de rúbrica:</b> La aplicación permite realizar la evaluación de un atestado por medio de una rúbrica	El usuario puede evaluar cada rúbrica de manera satisfactoria. Se verifica que todas las rúbricas fueron evaluadas.
<b>RQ013</b>	<b>Agregar observaciones a atestados:</b> El sistema permite adjuntar observaciones a las evaluaciones realizadas a un atestado	El usuario logra insertar las observaciones que considere necesarias a la hora de evaluar un atestado.
<b>RQ014</b>	<b>Tabla histórica:</b> El sistema genera una tabla histórica en el que se consideran los puntajes obtenidos en pasos anteriores.	El usuario logra revisar la tabla histórica para analizar el progreso de los funcionarios.
<b>RQ015</b>	<b>Generar tabla resumen:</b> La aplicación genera una tabla resumen en la que se incluye: nombre de los atestados, autores o participantes, punte obtenido, puntaje asignado según distribución, observaciones, nombre de documentos asociados, suma de puntos totales generales y suma de puntos totales según el artículo 77.	El usuario puede visualizar la información general de los atestados de manera simple y concisa.
<b>RQ016</b>	<b>Envío de evaluación por correo electrónico:</b> El sistema envía por medio de un correo electrónico el resultado de la evaluación realizada al funcionario correspondiente y a los revisores.	La información del resultado de la evaluación llega exitosamente al correo del funcionario para que pueda ser vista.
<b>RQ017</b>	<b>Verificación de puntajes:</b> El sistema verifica que el puntaje corresponde a la categoría correcta según el paso solicitado.	El usuario puede ver que no se asignen más puntos de los permitidos en determinado paso para ascenso a una categoría.
<b>RQ018</b>	<b>CRUD atestado:</b> CRUD para el atestado de Libro como usuario Revisor.	El usuario puede crear, editar, visualizar y eliminar atestados de tipo Libro de manera satisfactoria.
<b>RQ019</b>	<b>CRUD atestado:</b> CRUD para el atestado de Artículo como usuario Revisor.	El usuario puede crear, editar, visualizar y eliminar atestados de tipo Artículo de manera satisfactoria.

Código	Descripción	Criterios de Calidad
<b>RQ020</b>	<b>CRUD atestado:</b> CRUD para el atestado de Obra didáctica como usuario Revisor.	El usuario puede crear, editar, visualizar y eliminar atestados de tipo Obra didáctica de manera satisfactoria.
<b>RQ021</b>	<b>CRUD atestado:</b> CRUD para el atestado de Obra administrativa de desarrollo como usuario Revisor.	El usuario puede crear, editar, visualizar y eliminar atestados de tipo Obra administrativa de manera satisfactoria.
<b>RQ022</b>	<b>CRUD atestado:</b> CRUD para el atestado de Proyecto de investigación-extensión como usuario Revisor.	El usuario puede crear, editar, visualizar y eliminar atestados de tipo Proyecto de investigación-extensión de manera satisfactoria.
<b>RQ023</b>	<b>CRUD atestado:</b> CRUD para el atestado de Otras obras profesionales como usuario Revisor.	El usuario puede crear, editar, visualizar y eliminar atestados de tipo Otras obras profesionales de manera satisfactoria.
<b>RQ024</b>	<b>Evaluación de atestado:</b> Evaluación del atestado Libro como usuario Revisor.	El usuario puede evaluar correctamente atestados de tipo Libro.
<b>RQ025</b>	<b>Evaluación de atestado:</b> Evaluación del atestado de Artículo como usuario Revisor.	El usuario puede evaluar correctamente atestados de tipo Artículo.
<b>RQ026</b>	<b>Evaluación de atestado:</b> Evaluación del atestado de Obra didáctica como usuario Revisor.	El usuario puede evaluar correctamente atestados de tipo Obra didáctica.
<b>RQ027</b>	<b>Evaluación de atestado:</b> Evaluación del atestado de Obra administrativa de desarrollo como usuario Revisor.	El usuario puede evaluar correctamente atestados de tipo Obra administrativa.
<b>RQ028</b>	<b>Evaluación de atestado:</b> Evaluación del atestado de Proyecto de investigación-extensión como usuario Revisor.	El usuario puede evaluar correctamente atestados de tipo Proyecto de investigación-extensión.

## Capítulo 3

# Arquitectura del sistema

### 3.1. Diseño general del sistema

El sistema SIRA está diseñado utilizando la tecnología de WCF (Windows Communication Foundation). Este es un framework que permite desarrollar aplicaciones orientadas como servicio (Microsoft, 2021). Esta es la forma preferida de desarrollar aplicaciones del DATIC. En términos simples y concretos la forma en la que se está utilizando WCF en este proyecto es como un Modelo-Vista-Controlador con más capas (de negocios, utilitarias, Windows Service, etc.). Las capas adicionales tienen como propósito crear un estándar de comunicación entre las distintas aplicaciones desplegadas por el DATIC.

#### 3.1.1. Diseño de la persistencia

La persistencia de datos se logra mediante una base de datos relacional (MSSQL) la cual está desarrollada en conjunto con las clases modelo del sistema.

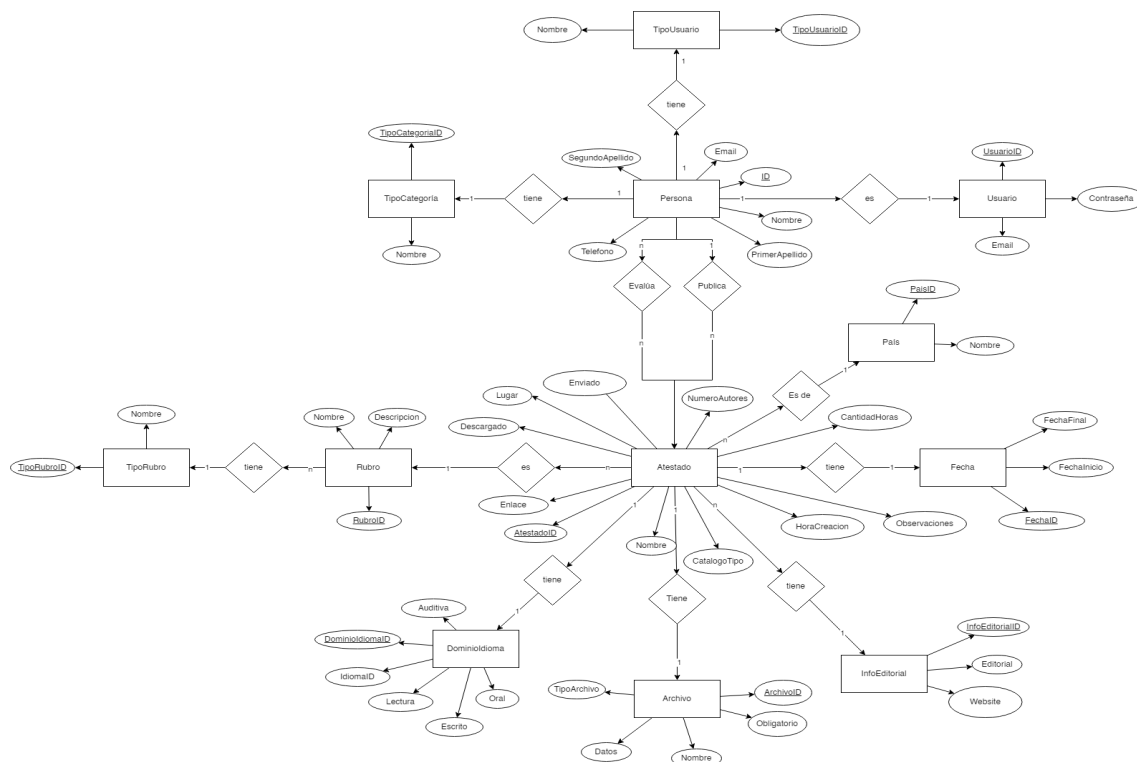


Figura 3.1: Diagrama entidad-relación de la base de datos del sistema.

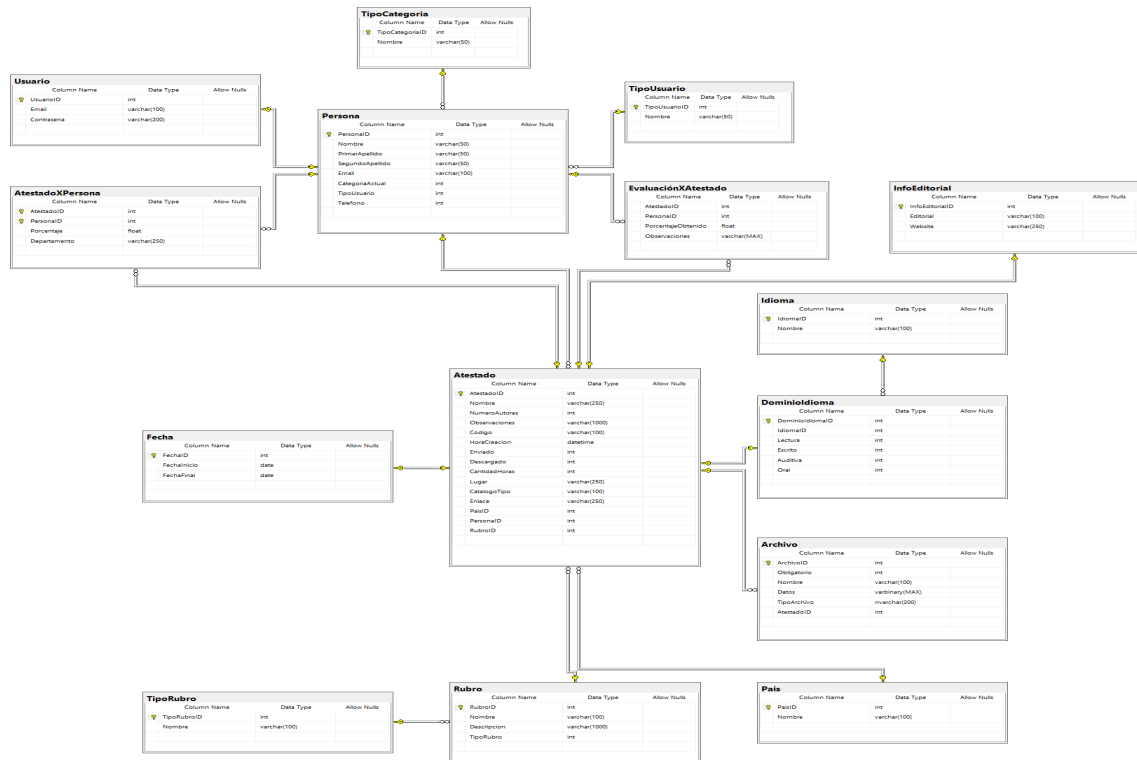


Figura 3.2: Diagrama relacional del sistema.

Para la base de datos se realizaron ciertos cambios para tomar en cuenta los nuevos requerimientos.

- Se agregó la tabla de TipoCategoría, esta contiene la categoría o prestigio que posee el docente. (Catedrático, por ejemplo)
- Se agregó la tabla TipoUsuario, la cual contiene el tipo de usuario. Existen 3, Administrador, Revisor, y Docente.
- Se agregó la tabla EvaluaciónXAtestado, esta contiene todas las notas y observaciones que dieron los distintos revisores a un atestado específico.

### 3.1.2. Diagrama de clases

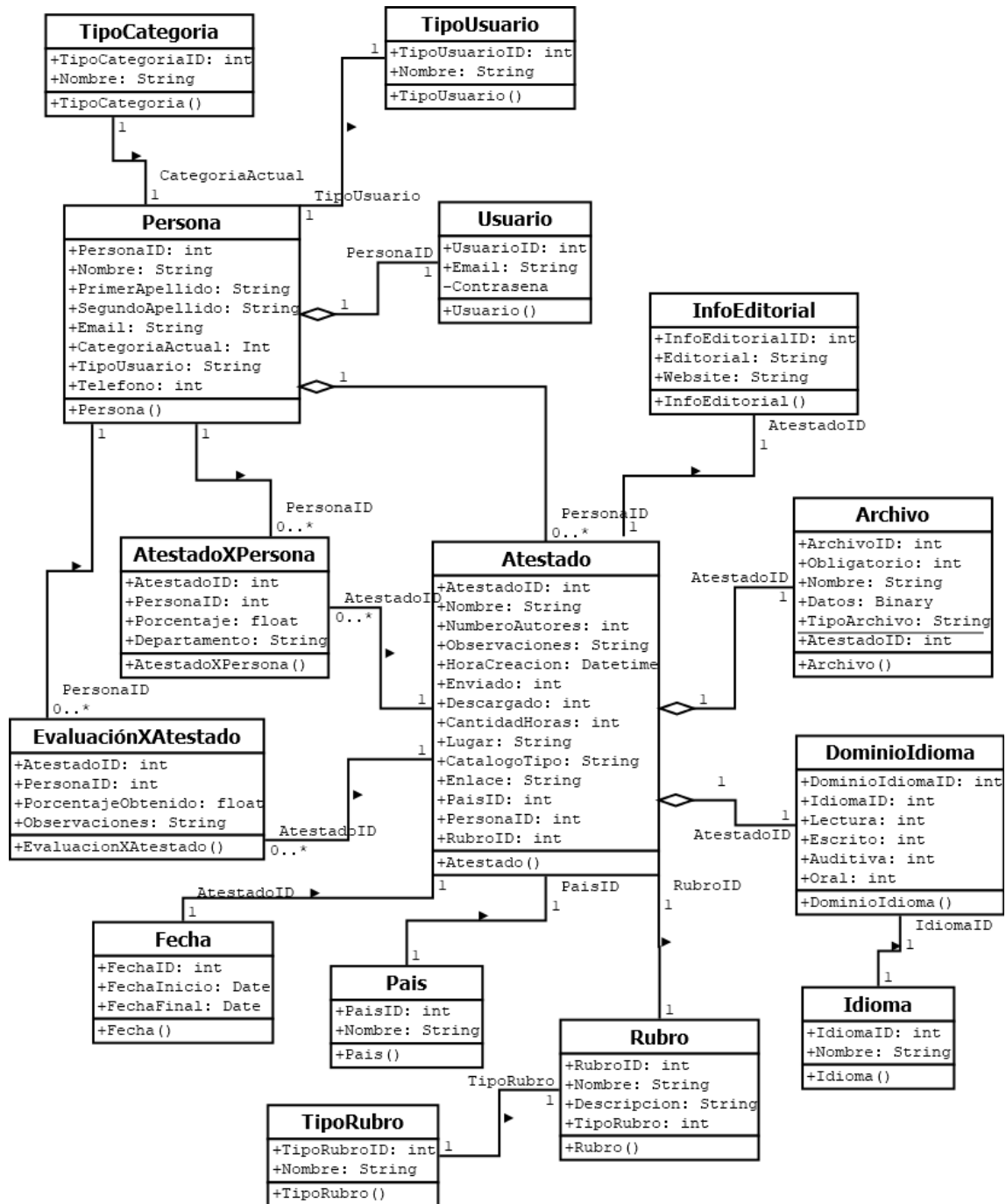


Figura 3.3: Diagrama de clases del sistema.

# Capítulo 4

## Instrucciones de Instalación

### 4.1. Entorno de Desarrollo y Producción

#### 4.1.1. Dependencias

- **IDE:** Visual Studio 2019/2022
- .NET Framework 4.8
- **Base de Datos:** Microsoft SQL Server 2019 15.0.2080.9 (X64)
- **Bibliotecas:**
  - Atestados.Datos
    - EntityFramework 6.0.0
  - Atestados.Negocios
    - AutoMapper 7.0.1
    - BCrypt.Net-Next 4.0.2
    - EntityFramework 6.1.3
    - System.ValueTuple 4.5.0
  - Atestados.Objetos
    - AutoMapper 7.0.1
    - EntityFramework 6.1.3
    - System.ValueTuple 4.5.0
  - Atestados.UI
    - Antlr 3.5.0.2
    - AutoMapper 7.0.1
    - EntityFramework 6.1.3
    - Bootstrap 5.1.3
    - JQuery 3.4.1
    - JQuery.UI.Combined 1.12.1
    - JQuery.Validation 1.17.1
    - Newtonsoft.Json 11.0.1
    - Owin 1.0.0
    - RestSharp 106.6.10
    - Respond 1.4.2

- WebGrease 1.6.0

Las instrucciones para instalar las dependencias se muestran en la sección 4.1.2.

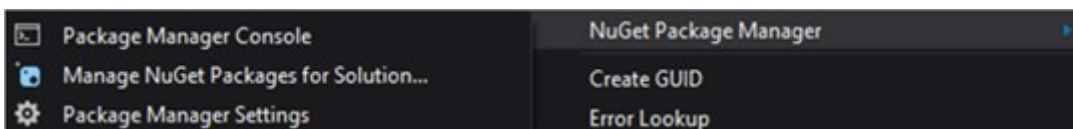
## 4.1.2. Instrucciones de instalación

### Base de Datos

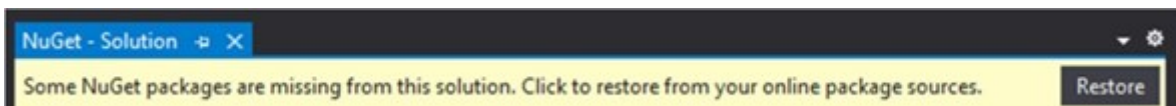
- Instalar SQL Server 2019
- (Opcional) Instalar SQL Server Management Studio 2019 para configurar la base de datos fácilmente
- Una vez que este corriendo el motor de SQL Server crear la base de datos con los scripts que se encuentran en la carpeta Base de Datos en el directorio de la solución. Cada script tiene un número el cual indica el orden en el que se deben de ejecutar, uno de estos scripts es para la creación del usuario administrador inicial cuyos datos de inicio son los siguientes email: admin@mail.com, contraseña: admin, si es necesario se puede crear otro usuario administrador una vez ya dentro del sistema y eliminar el rol de este.

### Solución

- Instalar Microsoft Visual Studio (la versión actual es la 2022)
- Mostrar los archivos ocultos en la carpeta de la solución, si existe una carpeta llamada .vs, eliminarla.
- Abrir la solución Atestados.sln en Visual Studio.
- Abrir NuGet Package Manager en Tools → NuGet Package Manager Manage NuGet Packages for Solution...



- Cuando esté abierto el administrador de paquetes va a salir un aviso el cual dice que algunos paquetes faltan, solo presione el botón de Restore para instalarlos



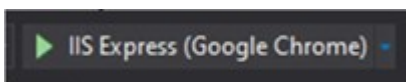
- (Para este paso es necesario que el motor de la base de datos este activo) Ir a View → Server Explorer. En este espacio, presione click derecho sobre las conexiones para agregar una nueva. En Data source debería de estar seleccionado SQL Server. En Server name les va a cargar el nombre de su servidor en la lista para seleccionar. En select database van a ingresar Atestados y presionan ok. Con esto ya debería estar creada la conexión.

- Después seleccionan la conexión y en la ventana de propiedades les va a mostrar el connection string. Copiándolo

(Name)	Atestados
<b>Connection</b>	
Connection String	Data Source=DESKTOP-D5JGAL7;Initial Catalog=
Provider	.NET Framework Data Provider for SQL Server
State	Open

- En Atestados.Datos hay un archivo llamado App.Config y en Atestados.UI hay uno llamado Web.config. Ambos tienen un nodo que se llama <connectionStrings>, dentro de este nodo hay uno que se llama add con name = "AtestadosEntities" y un atributo de este nodo es connectionString. En el connectionString deben buscar donde dice provider connection string y reemplazar lo que está dentro de &quot; con lo que obtuvieron en el paso anterior.
- Asegurarse que el Proyecto solo inicie desde Atestados.UI. Para hacerlo, en el explorador de la solución presione click derecho en la solución, luego propiedades y ahí asegúrese que este seleccionado Atestados.UI como proyecto de inicio.

- Para ejecutar la solución solo presione el botón de start o la tecla F5



### 4.1.3. Acceso al Servidor

- Para ingresar al sistema desde el servidor es necesario tener acceso a la VPN del TEC, esto se debe coordinar con el DATIC.
- El acceso al servidor es por medio del acceso remoto de Windows con los siguientes datos: IP: 172.19.127.254, Usuario: Administrator, Contraseña: Rodolfo\_Mora.
- Si solo se desea ingresar a la página web del sistema sería utilizando la dirección IP mencionada anteriormente en un navegador web, es necesario tener la VPN del TEC para poder realizar esto. El usuario administrador es admin@mail.com y su contraseña es admin.



- Cuando obtenga un usuario para la VPN del TEC, lo más probable es que tenga que utilizar el número de carné como contraseña en lugar de la contraseña en sí.

# Capítulo 5

## Métodos del API

En esta sección se detallan los métodos que se tienen publicados en la sección del backend del sistema. Por cada método se incluyen parámetros necesarios, formato de envío, y los procedimientos necesarios para comprender y utilizar correctamente el API.

- /Articulo/getAutores

- type: POST
- contentType: application/json; charset=utf-8

- /Funcionario/UsuarioPorEmail

- type: POST
- async: false
- contentType: application/json; charset=utf-8
- dataType: json
- data: JSON en formato String con la información del usuario

- /Articulo/AgregarAutor

- type: POST
- contentType: application/json; charset=utf-8
- dataType: json
- data: JSON en formato String con la información del autor

- /Articulo/borrarAutor

- type: POST
- contentType: application/json; charset=utf-8
- dataType: json
- data: JSON en formato String con la información del autor

- /Libro/Cargar

Este método se puede aplicar de igual manera para otros tipos de atestados, solo habría que cambiar el url por el nombre de atestado correspondiente.

- type: POST
- contentType: false

- dataType: json
- data: FormData
- cache: false
- processData: false
- /Funcionario/notaAtestado
  - type: POST
  - contentType: application/json; charset=utf-8
  - dataType: json
  - data: JSON en formato String con un atributo AtestadoID
- /Atestados/AsignarPuntos
  - type: POST
  - contentType: application/json; charset=utf-8
  - dataType: json
  - data: JSON de autores en formato String
- /Atestados/ObtenerAutores
  - type: POST
  - contentType: application/json; charset=utf-8
- /Login/ConsultarPermisosUsuario
  - type: POST
  - contentType: application/json; charset=utf-8
  - data: JSON con el atributo id

## Capítulo 6

# Conclusiones y trabajo futuro

### 6.1. Conclusiones

Durante el desarrollo del proyecto se adquirieron una serie de lecciones que nos permitieron virar el curso del proyecto hacia lo que considerábamos que era el camino correcto. Entre estas conclusiones se encuentra que: al momento de utilizar metodologías de desarrollo ágil es importante asignar desde un inicio las tareas que va a realizar cada integrante para llevar un orden el proceso de integración continua y desarrollo continuo. A su vez se aprendió que al ser este un proyecto heredado, hay que tomar en consideración el tiempo que implica comprender la totalidad de lo desarrollado y la corrección de errores como parte de las tareas del proyecto. En última instancia, se considera de gran importancia mantener una relación cercana con el cliente para que el proceso de desarrollo del producto se realice de la manera más transparente y eficiente posible.

### 6.2. Problemáticas y limitaciones

Al ser un proyecto en desarrollo, es natural que se presenten numerosas limitaciones. Al inicio de este trabajo, el proyecto, que fue desarrollado por otro grupo anteriormente, no contaba con considerables funciones básicas. La interfaz era poco intuitiva e incómoda de usar, no se podían evaluar atestados, la base de datos no tomaba en cuenta ciertos tipos de usuario, entre otros.

En el estado actual del desarrollo, todavía hay muchas limitaciones debido a que el trabajo requerido para concluir el proyecto que es SIRA es considerable, y muy probablemente no se termine en esta iteración. Entre estas limitaciones se incluye la falta de listas de verificación para la evaluación de los atestados, no cuenta con una barra de progreso para que el revisor pueda visualizar la categoría de un funcionario, no se puede transferir un atestado de una categoría a otra, entre otros.

### 6.3. Trabajo futuro

En el futuro habrá mucho trabajo por hacer en este proyecto. La evaluación por medio de listas de verificación debería ser una prioridad. También corregir ciertos aspectos del diseño gráfico y la interfaz que necesitan más atención, pero que no han sido prioridad en esta iteración. Además, falta el envío por medio de correo electrónico a los funcionarios cuando su atestado ha sido evaluado.

Sin embargo, que espera que agregar estas funcionalidades no sea tan difícil, ya que se ha trabajado en tener una base sólida de modelos y clases a sobre las cuales se puede trabajar y extender el proyecto.

## Referencias bibliográficas

Microsoft. (2021, Dec). *What is windows communication foundation*. <https://docs.microsoft.com/en-us/dotnet/framework/wcf/whats-wcf>.