|  |  |
| --- | --- |
|  | Descripción breve  La academia ALWAYS IDIOMAS solicita un software el cual les permita mantener y controlar la gestión administrativa.  bryan gtt |

Contenido

[1 Introducción. 3](#_Toc507273160)

[2 Objetivo. 3](#_Toc507273161)

[3 Análisis. 3](#_Toc507273162)

[3.1 ASI 1: Definición del sistema. 3](#_Toc507273163)

[3.1.1 ASI 1.1: Determinación del alcance del sistema. 3](#_Toc507273164)

[3.1.2 ASI 1.3: Especificación de estándares y normas. 3](#_Toc507273165)

[3.2 ASI 2: Establecimiento de requisitos. 4](#_Toc507273166)

[3.2.1 ASI 2.1: Obtención de requisitos. 4](#_Toc507273167)

[3.2.2 IEEE830 2: Descripción general. 4](#_Toc507273168)

[3.2.3 IEEE830 2.2: Funcionalidades del producto. 4](#_Toc507273169)

[3.2.4 IEEE830 2.3: Características de los usuarios. 4](#_Toc507273170)

[3.2.5 IEEE830 2.4: Restricciones. 4](#_Toc507273171)

[3.2.6 IEEE830 3: Requisitos específicos. 4](#_Toc507273172)

[3.2.7 IEEE830 3.1 Interfaces Externas. 4](#_Toc507273173)

[3.2.7.1.1 Interfaz Inicial. 5](#_Toc507273174)

[3.2.7.1.2 Interfaz de Login. 5](#_Toc507273175)

[3.2.7.1.3 Interfaz gestión de Alumnos. 5](#_Toc507273176)

[3.2.7.1.4 Interfaz de gestión de profesores y alumnos. 5](#_Toc507273177)

[3.2.7.1.5 Interfaz para añadir y modificar usuarios o empleados. 6](#_Toc507273178)

[3.2.7.1.6 Interfaz para asignar bonos (clases). 6](#_Toc507273179)

[3.2.7.1.7 Interfaz para asignar Tarifas especiales 7](#_Toc507273180)

[3.2.7.1.8 Interfaz de información sobre la empresa. 7](#_Toc507273181)

[3.2.7.1.9 Interfaz del informe de cada alumno nuevo. 7](#_Toc507273182)

[3.2.7.1.10 Interfaz para el informe de empleados. 8](#_Toc507273183)

[3.2.7.1.11 Interfaz para el informe de alumnos. 8](#_Toc507273184)

[3.2.8 ASI 2.1: Especificación de casos de uso. 9](#_Toc507273185)

[3.2.9 CUID\_01: Gestión empleados. 9](#_Toc507273186)

[3.2.10 CUID\_02: Gestión de alumnos. 10](#_Toc507273187)

[3.3 ASI 5: Análisis de clases. 11](#_Toc507273188)

[3.4 ASI 6: Elaboración del modelo de datos 12](#_Toc507273189)

[3.5 ASI 8: Definición de interfaces de usuario 13](#_Toc507273190)

[3.6 ASI 8.3 Especificación de los formatos individuales de la interfaz de pantalla. 13](#_Toc507273191)

[3.7 ASI 8.4 Especificación de comportamiento dinámico de la interfaz. 13](#_Toc507273192)

[3.8 ASI 10: Especificación del plan de pruebas. 13](#_Toc507273193)

[4 Diseño. 13](#_Toc507273194)

[4.1 DSI 1.6: Especificación del entorno tecnológico. 13](#_Toc507273195)

[4.2 DSI 4: Diseño de clases. 13](#_Toc507273196)

[4.2.1 DSI 3.1 Identificación de clases asociadas a casos de uso. 13](#_Toc507273197)

[5 Glosario. 15](#_Toc507273198)

[6 Web grafía. 16](#_Toc507273199)

# Introducción.

En este documentos se detallan las fases de análisis, diseño y construcción por las cuales se han pasado a la hora de desarrollar un software de gestión destinado a una academia de inglés.

Para la distribución de apartados y definiciones se ha decidido adaptar a nuestro propósito la Metodología de Planificación, Desarrollo y Mantenimiento de sistemas de información Métrica V3 y el estándar de especificación de requisitos del software IEEE830, así como también se hace uso de un lenguaje de modelado de software (UML).

# Objetivo.

Se persiguen varios objetivos en este documento:

* Proporcionar a los programadores una idea clara de qué hay que hacer y cómo.
* Proporcionar a aquel que no esté relacionado con el mundo de la informática información sobre qué hará el software y cómo.

# Análisis.

## ASI 1: Definición del sistema.

### ASI 1.1: Determinación del alcance del sistema.

Se pretende desarrollar un software el cual permita a los trabajadores de una academia de inglés mantener y controlar la gestión administrativa de empleados, alumnos y clases que se imparten.

Por cada nuevo alumno se generará un informe que tendrá la apariencia definida en el *punto* ***X.*** de este documento, así como también se generarán informes de los alumnos y empleados actuales con la apariencia definida en el *punto* ***X*** *de este documento.*

El cliente solicita también la posibilidad de asignar bonos y tarifas especiales a sus alumnos, pues la academia ofrece un peculiar servicio que consiste en descuentos por grupos, por horas de asistencia y variedad de precios según las clases que se impartan en verano, invierno, de manera presencial, a distancia… para más detalles sobre este peculiar servicio ver el p*unto* ***X*** de este documento.

### ASI 1.3: Especificación de estándares y normas.

Normas a tener en cuenta y estándares a utilizar.

* Dado que este software trata información personal se ha de tener en cuenta la Ley Orgánica 15/1999, de 13 de diciembre de protección de datos.
* Dado que la aplicación se implementará dentro del territorio español los carnés identificativos de las personas deben constar de nueve dígitos: ocho numéricos seguidos de una letra.
* Dado que la aplicación se implementará dentro del territorio europeo la moneda utilizada será el EURO [€].

## ASI 2: Establecimiento de requisitos.

La información contenida en este apartado complementa los requisitos solicitados por el cliente.

### ASI 2.1: Obtención de requisitos.

A continuación se describen las funcionalidades con las cuales contará y NO contará el software.

#### IEEE830 2: Descripción general.

A continuación se definen las funcionalidades y restricciones.

#### IEEE830 2.2: Funcionalidades del producto.

1. Gestionar empleados: alta, baja, modificación.
2. Gestionar alumnos: alta baja, modificación, asignación de bonos (clases).
3. Gestionar Clases: permitir asociar clases (bonos) a los alumnos.
4. Permitir asociar alumnos a profesores.
5. Se debe permitir que el sistema genere informes de empleados, *apartado* ***X****.*
6. Se debe permitir que el sistema genere informes de alumnos, *apartado* ***X****.*
7. Se debe permitir que el sistema genere un informe por cada alumno nuevo, este informe debe contener la estructura definida en el *punto* ***X***

#### IEEE830 2.3: Características de los usuarios.

A continuación se definen las operaciones que podrán realizar los usuarios que acceden al sistema.

1. Los empleados deben tener una cuenta de usuario y una contraseña.
2. Se debe permitir a los profesores ver los alumnos de la academia.
3. Se debe permitir a quien haga de jefe ver los alumnos y los empleados de la academia.

#### IEEE830 2.4: Restricciones.

A continuación se definen las características de las que dispone el sistema así como también la limitación de los usuarios respecto a las operaciones que realiza cada uno.

1. Los profesores no pueden ver la lista de empleados, solo sus propios

alumnos y no los de otro profesor.

1. Aquel que sea jefe puede ver y gestionar cualquier alumno y cualquier empleado.
2. Los empleados que no sean ni jefes ni profesores no tienen acceso al sistema.

#### IEEE830 3: Requisitos específicos.

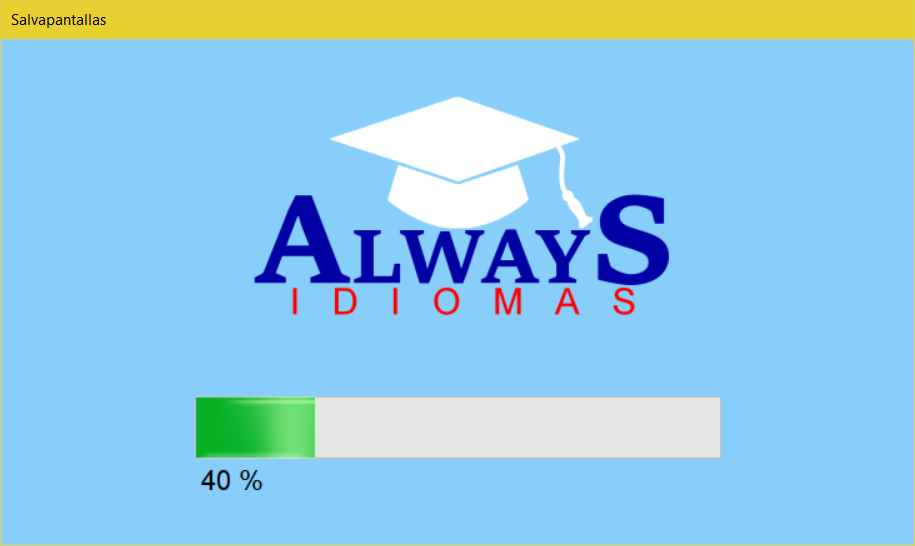
A continuación se definen las características de la interfaz de usuario.

#### IEEE830 3.1 Interfaces Externas.

El software dispondrá de un total de 11 interfaces, a continuación se presentan los prototipos de las mismas.

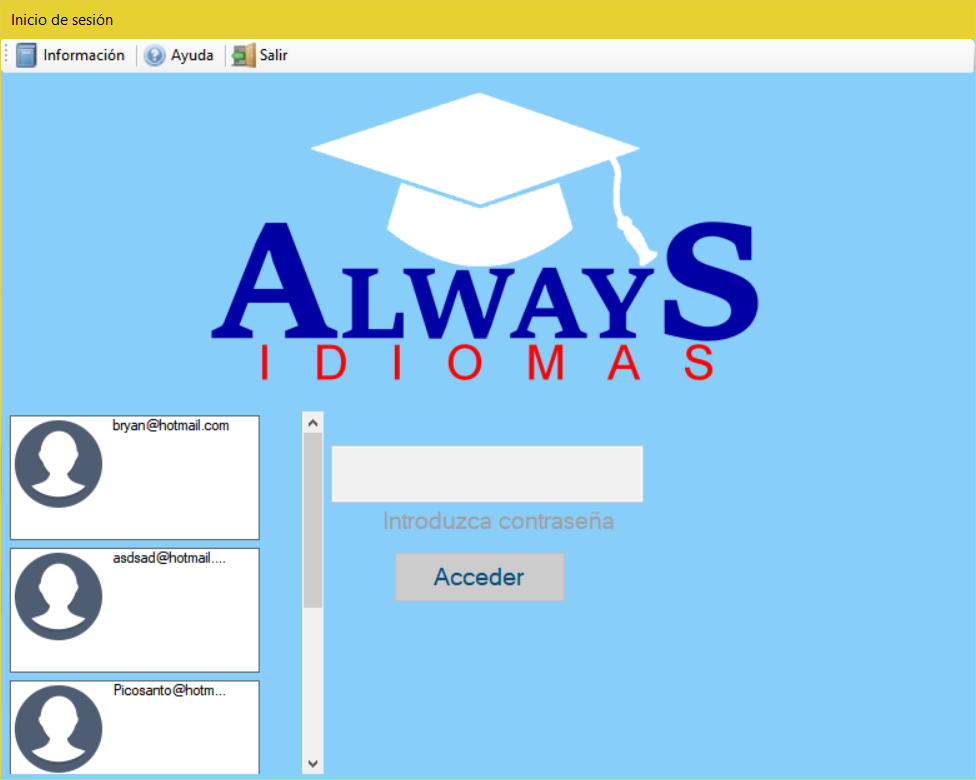
##### Interfaz Inicial.

Esta interfaz será lo primero que el usuario vea.



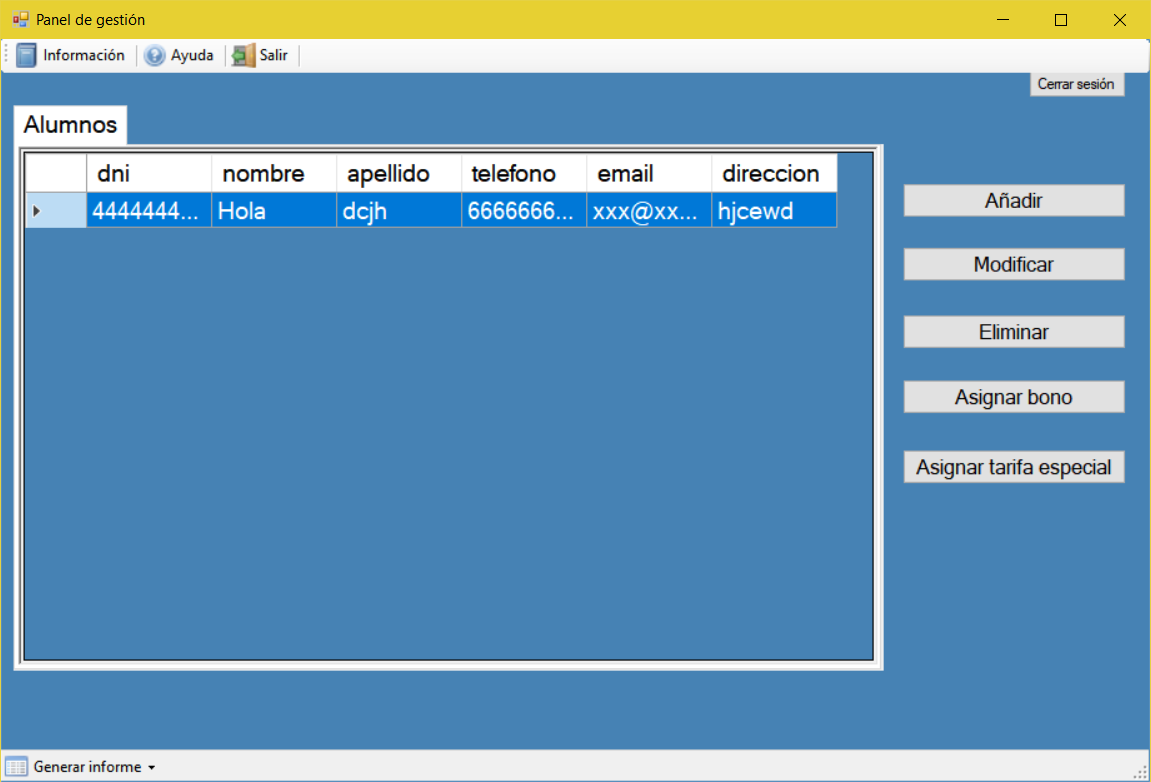
##### Interfaz de Login.

Interfaz que permite a los usuarios acceder al sistema.



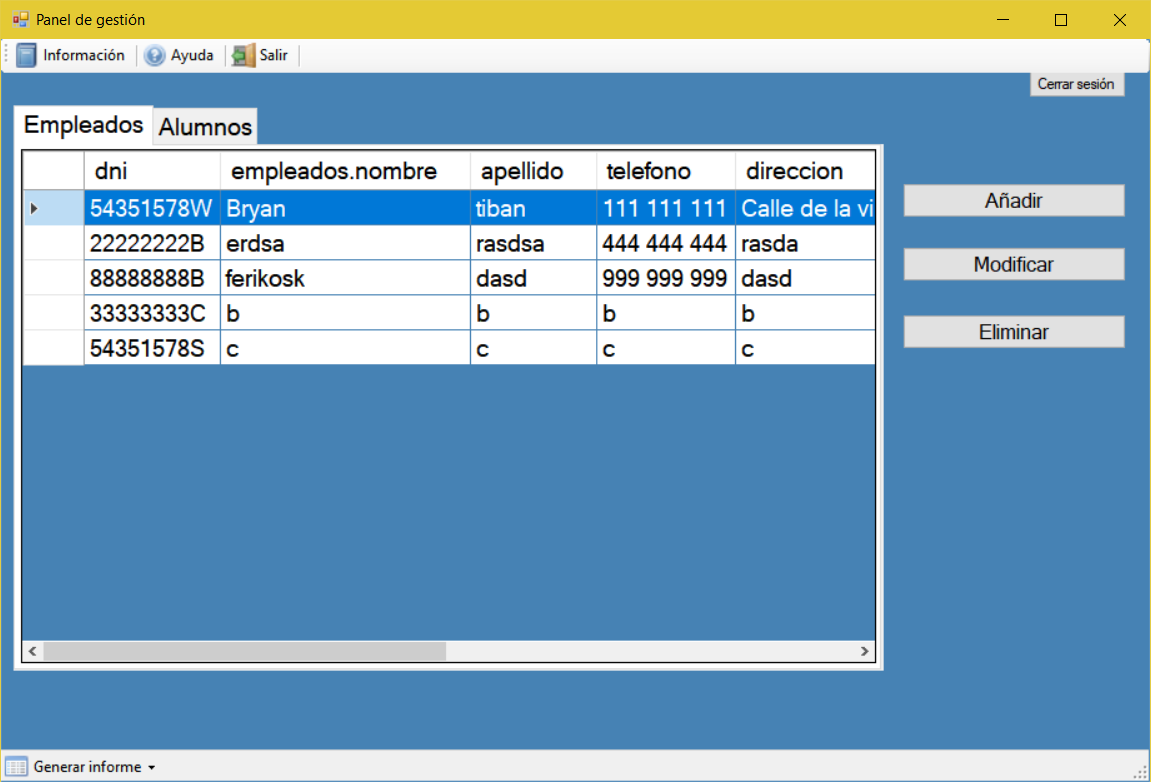
##### Interfaz gestión de Alumnos.

Esta interfaz la verán los profesores.



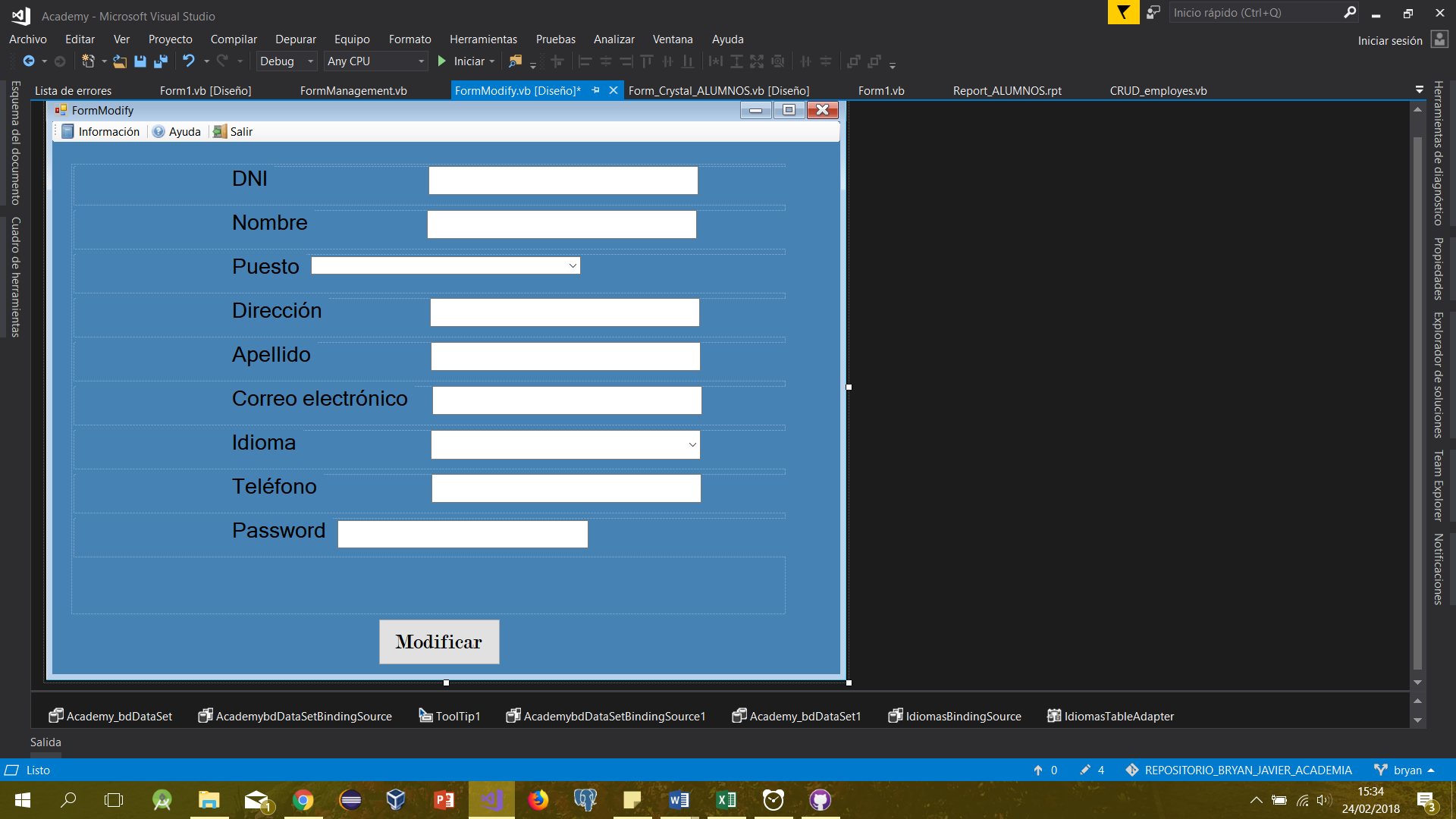
##### Interfaz de gestión de profesores y alumnos.

Esta interfaz la verán exclusivamente aquel o aquellos que sean jefes.



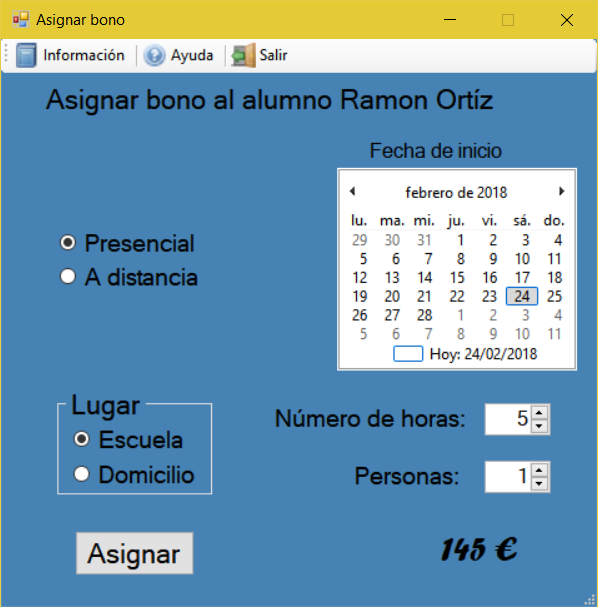
##### Interfaz para añadir y modificar usuarios o empleados.

Esta interfaz servirá para añadir nuevos empleados o nuevos alumnos.



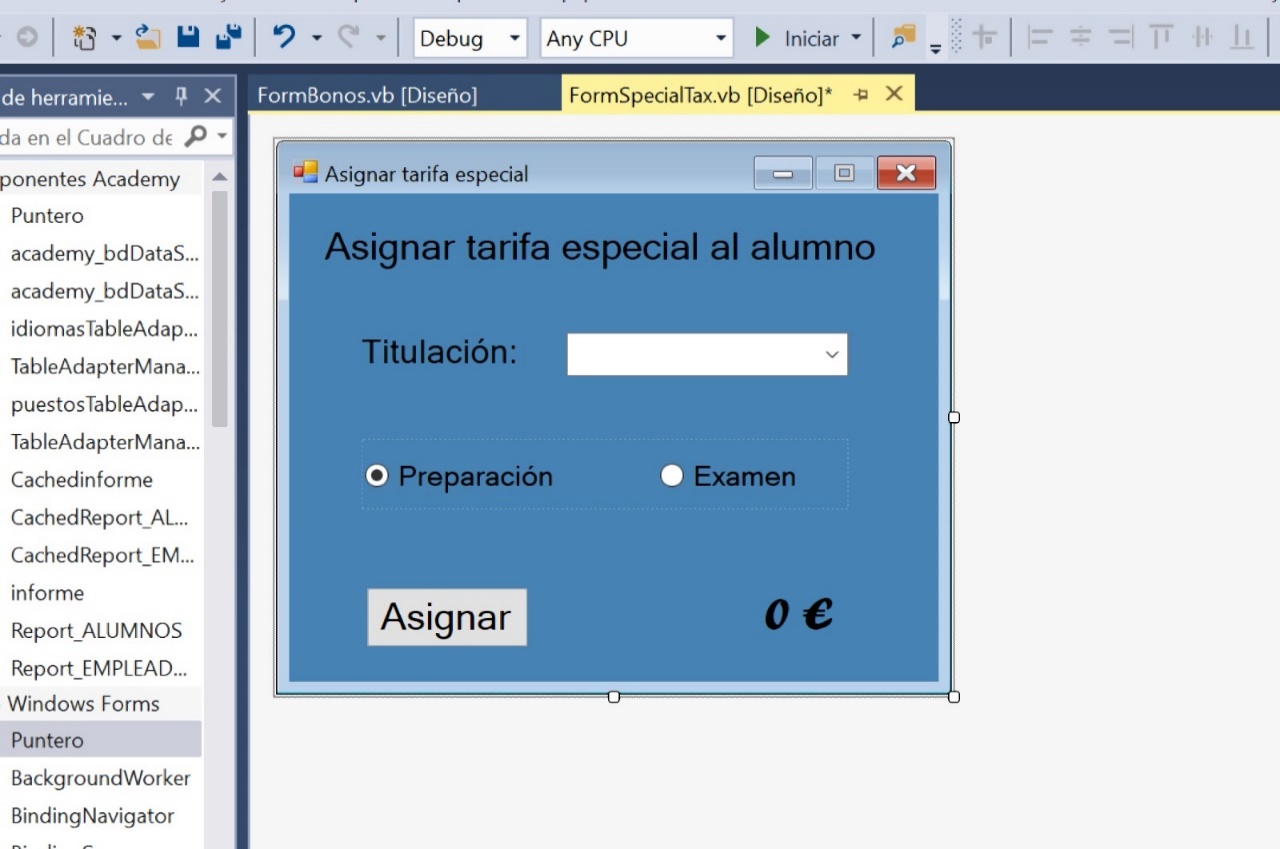
##### Interfaz para asignar bonos (clases).

Esta interfaz servirá para asignar bonos a los alumnos (véase punto **X** de este documento para mejor entendimiento sobre los bonos)

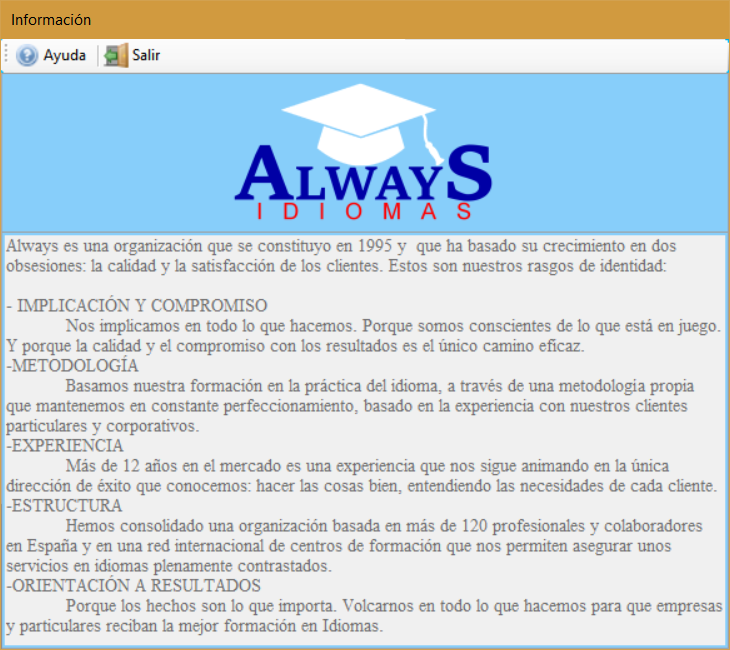


##### Interfaz para asignar Tarifas especiales

Esta interfaz servirá para asignar tarifas especiales a los alumnos (véase punto **X** de este documento para mejor entendimiento sobre las tarifas especiales).

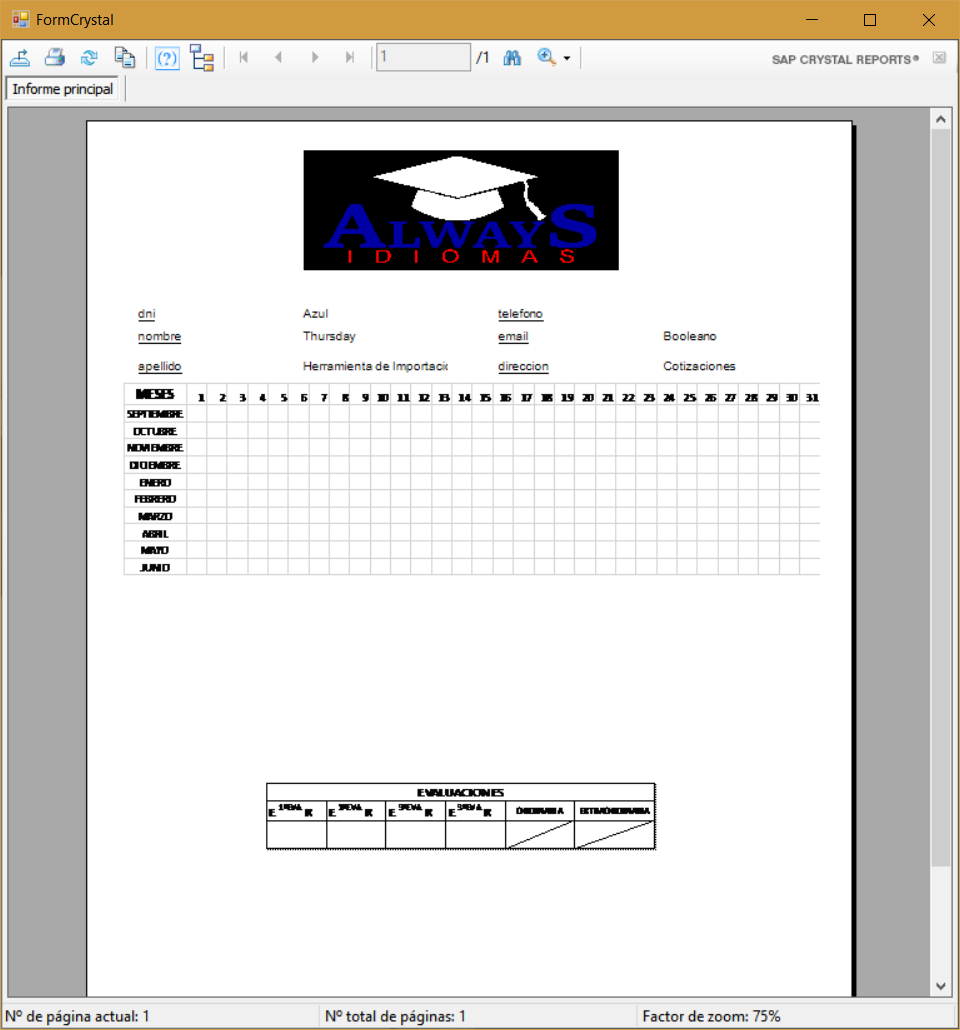


##### Interfaz de información sobre la empresa.



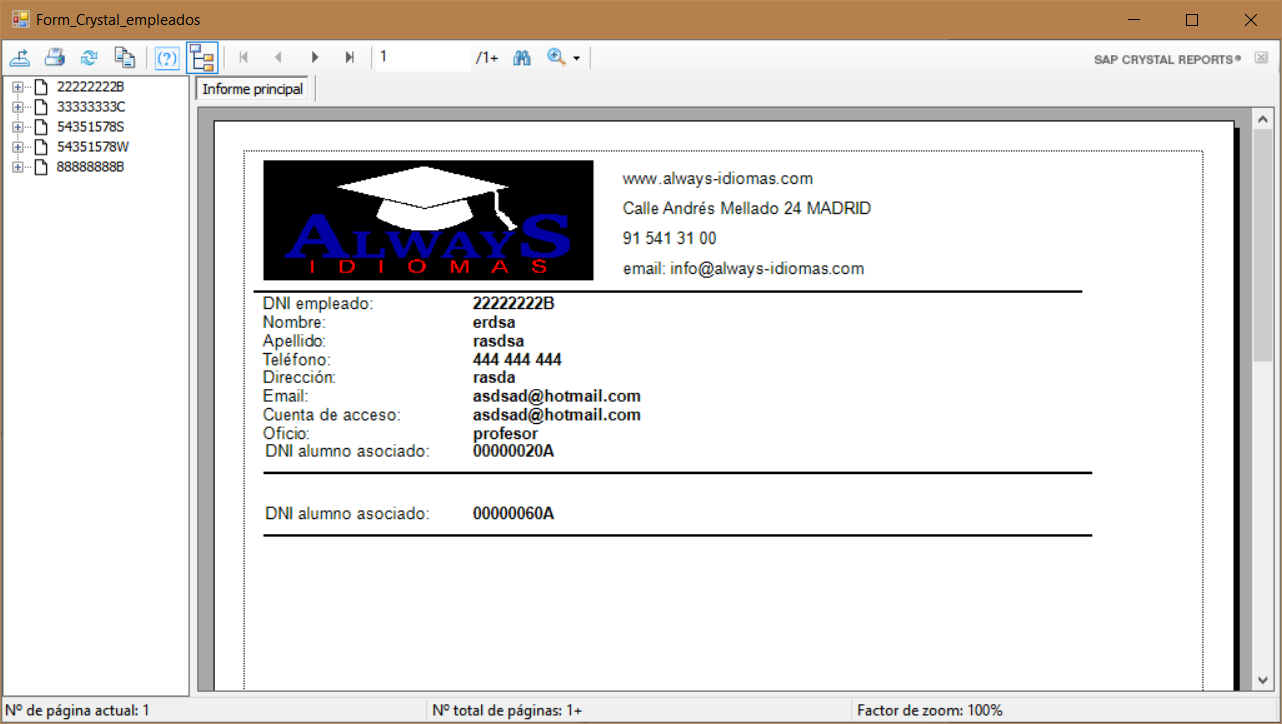
##### Interfaz del informe de cada alumno nuevo.

Esta interfaz se genera por cada nuevo alumno.



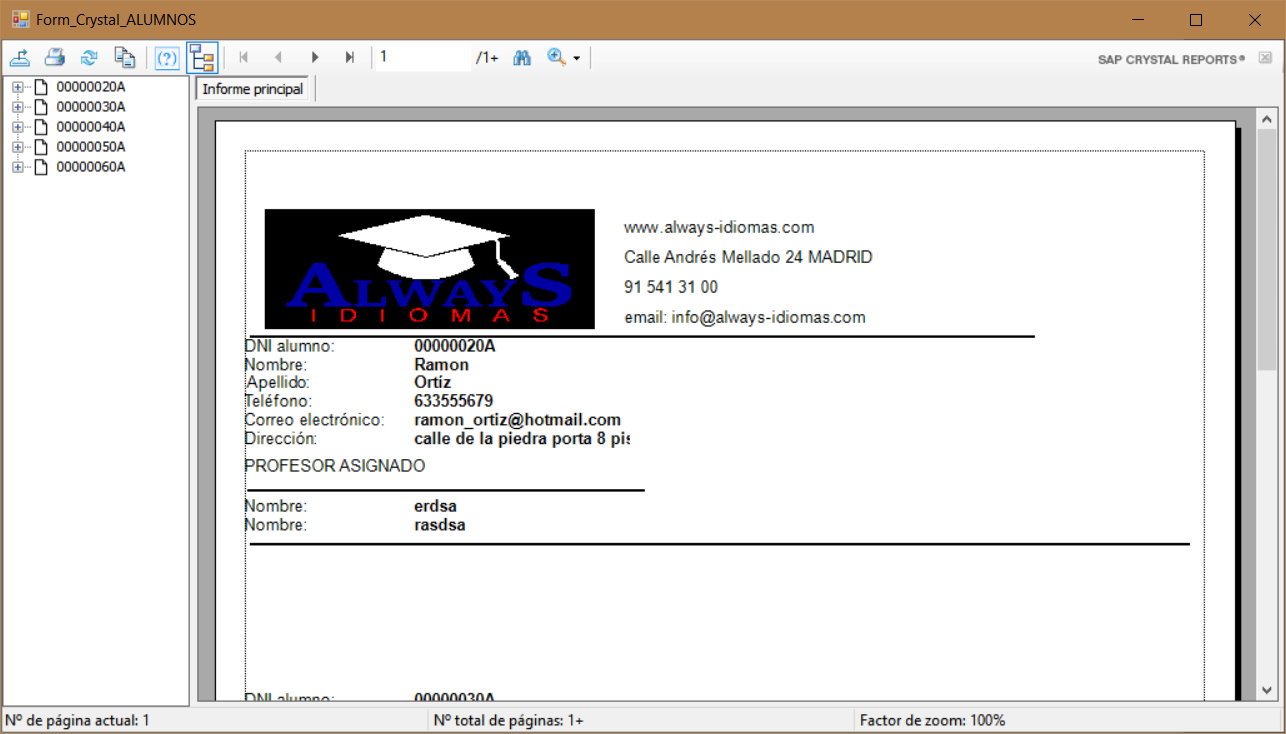
##### Interfaz para el informe de empleados.

Esta interfaz muestra toda la plantilla de trabajadores.

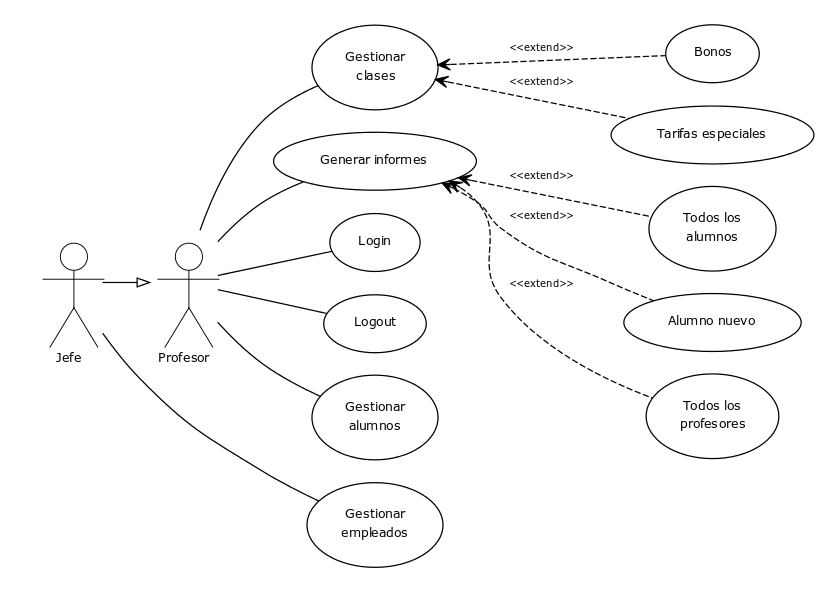


##### Interfaz para el informe de alumnos.

Esta interfaz muestra a todos los alumnos de la academia.



#### ASI 2.1: Especificación de casos de uso.

A continuación se presenta un diagrama el cual representa las funcionalidades del sistema 

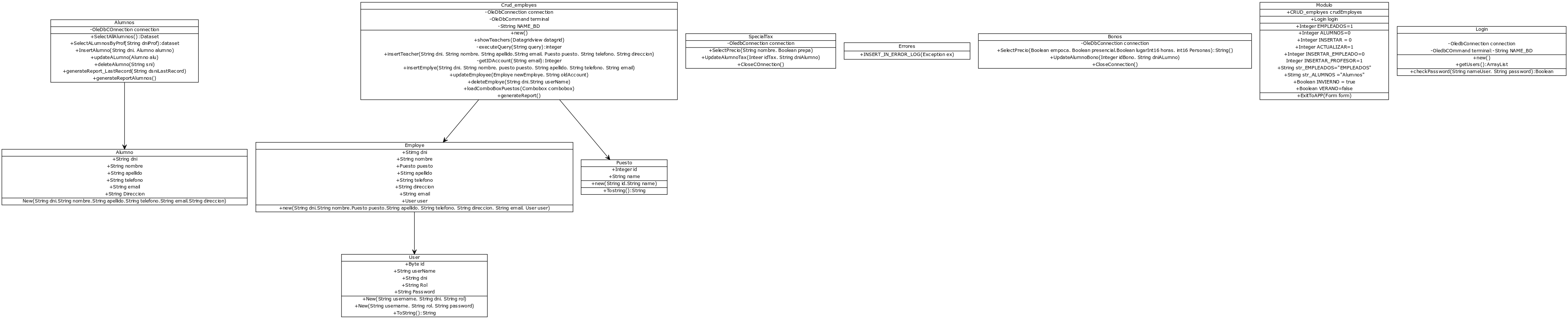
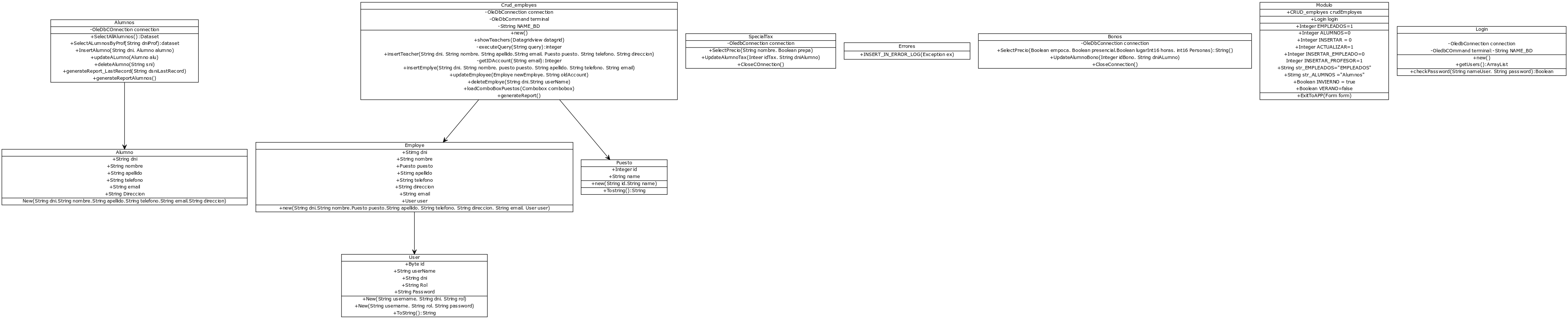
#### CUID\_01: Gestión empleados.

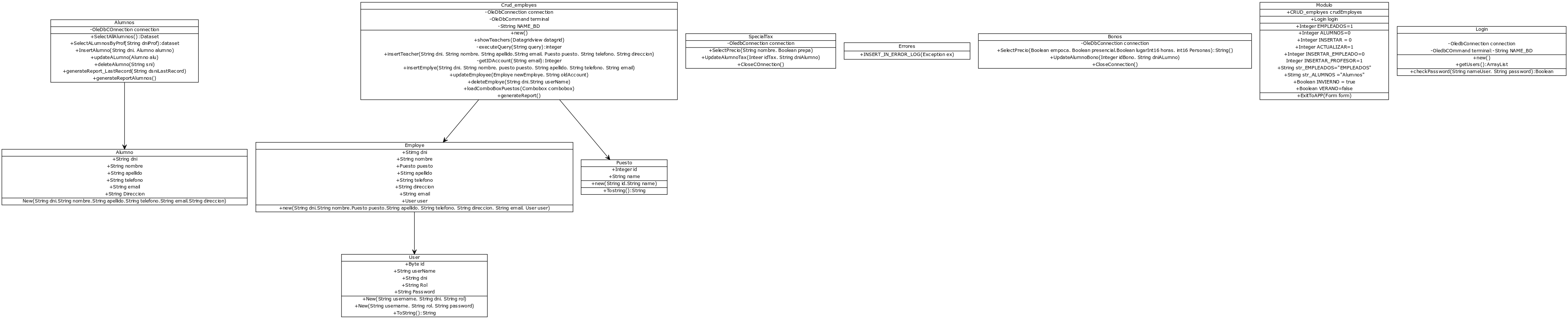
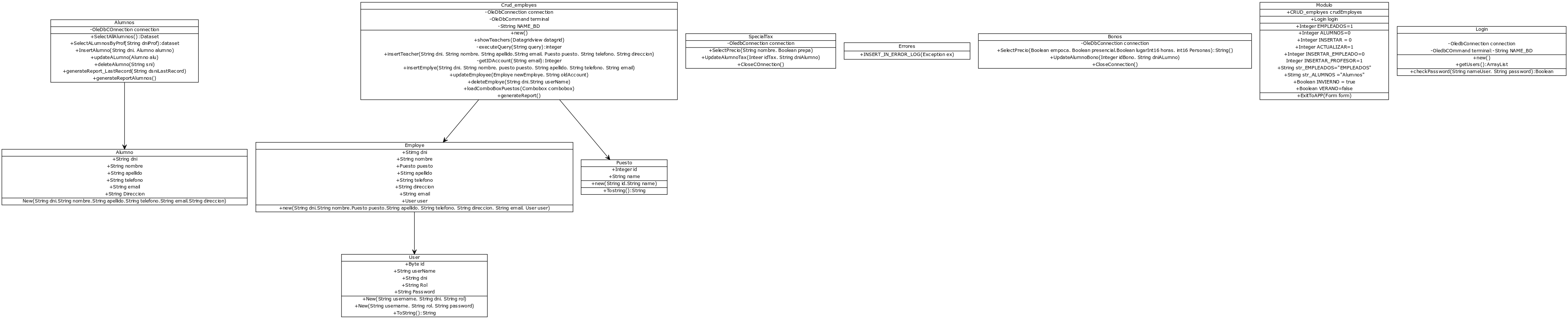
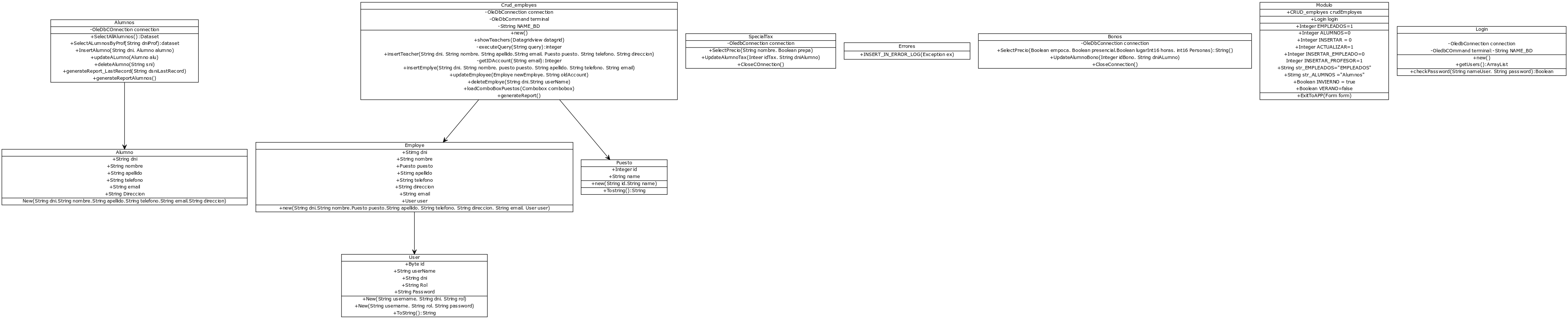
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DATOS GENERALES | | | | | | | | | |
| Autor | | Bryan Tibán | | Fecha | | 24/01/2018 | | Versión | 1.0 |
| Aplicación | | Always Academy | | | | | | | |
| Identificador | | CUID\_01: Gestión empleados. | | | | | | | |
| Prioridad | | Alta | Urgencia | | | | Alta | | |
| DESCRIPCIÓN BREVE \* | | | | | | | | | |
| Realización de altas bajas y modificaciones sobre los datos de los empleados. | | | | | | | | | |
| ACTORES \* | | | | | | | | | |
| * Jefe | | | | | | | | | |
| TRIGGER (DISPARADOR) | | | | | | | | | |
| Introducir datos referentes al empleado.   * DNI * Nombre * Cuenta * Puesto | | | | | * Apellido * Teléfono * Dirección * Email | | | | |
| PRECONDICIONES \* | | | | | | | | | |
| * Condición 1: verificar que DNI tenga 8 números seguidos de una letra. * Condición 2: verificar que el puesto asignado exista. * Condición 3: verificar que email tiene la sintaxis correcta (caracteres@caracteres.caracteres) * Condición 4: verificar que el teléfono contiene 9 dígitos. | | | | | | | | | |
| FLUJO PRINCIPAL o SECUENCIA NORMAL | | | | | | | | | |
| Paso | Acción | | | | | | | | |
| 1 | Recoger los datos que permiten la funcionalidad de A/B/M[[1]](#footnote-1) | | | | | | | | |
| 2 | Validaciones formales de los datos (indicado en las precondiciones). | | | | | | | | |
| 3 | Validaciones de negocio de los datos. | | | | | | | | |
| 4 | El sistema graba los datos en la base de datos. | | | | | | | | |
| 5 | Informar al usuario sobre el cambio realizado. | | | | | | | | |
| FLUJO ALTERNATIVO o Excepciones \* | | | | | | | | | |
| Paso | Acción | | | | | | | | |
| 2.1 | Los datos no se ajustan a las validaciones formales. | | | | | | | | |
| 3.1 | Los datos no se ajustan a las validaciones de negocio. | | | | | | | | |
| 4.1 | Dato duplicado/ Dato inexistente. | | | | | | | | |
| POSTCONDICIONES | | | | | | | | | |
| No aplica. | | | | | | | | | |
| EXTENSIONES | | | | | | | | | |
| No aplica. | | | | | | | | | |
| INCLUSIONES | | | | | | | | | |
| No aplica. | | | | | | | | | |
| Rendimiento | | | | | | | | | |
| No aplica. | | | | | | | | | |
| Frecuencia esperada | | | | | | | | | |
| No aplica. | | | | | | | | | |
| Volumen estimado | | | | | | | | | |
| Un único acceso, sin simultaneidad. | | | | | | | | | |
| Comentarios adicionales | | | | | | | | | |
| Operaciones A/B/M de los empleados. | | | | | | | | | |

#### CUID\_02: Gestión de alumnos.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DATOS GENERALES | | | | | | | | | |
| Autor | | Javier Hurtado | | Fecha | | 24/01/2018 | | Versión | 1.0 |
| Aplicación | | Always Academy | | | | | | | |
| Identificador | | CUID\_02: Gestión alumnos. | | | | | | | |
| Prioridad | | Alta | Urgencia | | | | Alta | | |
| DESCRIPCIÓN BREVE \* | | | | | | | | | |
| Realización de altas bajas y modificaciones sobre los datos de los alumnos. | | | | | | | | | |
| ACTORES \* | | | | | | | | | |
| * Jefe o Profesor | | | | | | | | | |
| TRIGGER (DISPARADOR) | | | | | | | | | |
| Introducir datos referentes al alumno.   * DNI * Nombre * Apellido | | | | | * Teléfono * Dirección * Email | | | | |
| PRECONDICIONES \* | | | | | | | | | |
| * Condición 1: verificar que DNI tenga 8 números seguidos de una letra. * Condición 2: verificar que el puesto asignado exista. * Condición 3: verificar que email tiene la sintaxis correcta (caracteres@caracteres.caracteres) * Condición 4: verificar que el teléfono contiene 9 dígitos. | | | | | | | | | |
| FLUJO PRINCIPAL o SECUENCIA NORMAL | | | | | | | | | |
| Paso | Acción | | | | | | | | |
| 1 | Recoger los datos que permiten la funcionalidad de A/B/M[[2]](#footnote-2) | | | | | | | | |
| 2 | Validaciones formales de los datos. | | | | | | | | |
| 3 | Validaciones de negocio de los datos (indicado en las precondiciones). | | | | | | | | |
| 4 | El sistema graba los datos en la base de datos. | | | | | | | | |
| 5 | Informar al usuario sobre el cambio realizado. | | | | | | | | |
| FLUJO ALTERNATIVO o Excepciones \* | | | | | | | | | |
| Paso | Acción | | | | | | | | |
| 2.1 | Los datos no se ajustan a las validaciones formales. | | | | | | | | |
| 3.1 | Los datos no se ajustan a las validaciones de negocio. | | | | | | | | |
| 4.1 | Dato duplicado/ Dato inexistente. | | | | | | | | |
| POSTCONDICIONES | | | | | | | | | |
| No aplica. | | | | | | | | | |
| EXTENSIONES | | | | | | | | | |
| No aplica. | | | | | | | | | |
| INCLUSIONES | | | | | | | | | |
| No aplica. | | | | | | | | | |
| Rendimiento | | | | | | | | | |
| No aplica. | | | | | | | | | |
| Frecuencia esperada | | | | | | | | | |
| No aplica. | | | | | | | | | |
| Volumen estimado | | | | | | | | | |
| Un único acceso, sin simultaneidad. | | | | | | | | | |
| Comentarios adicionales | | | | | | | | | |
| Operaciones A/B/M de los empleados. | | | | | | | | | |

## https://yuml.me/f2d4532c.pnghttps://yuml.me/f2d4532c.pngASI 5: Análisis de clases.

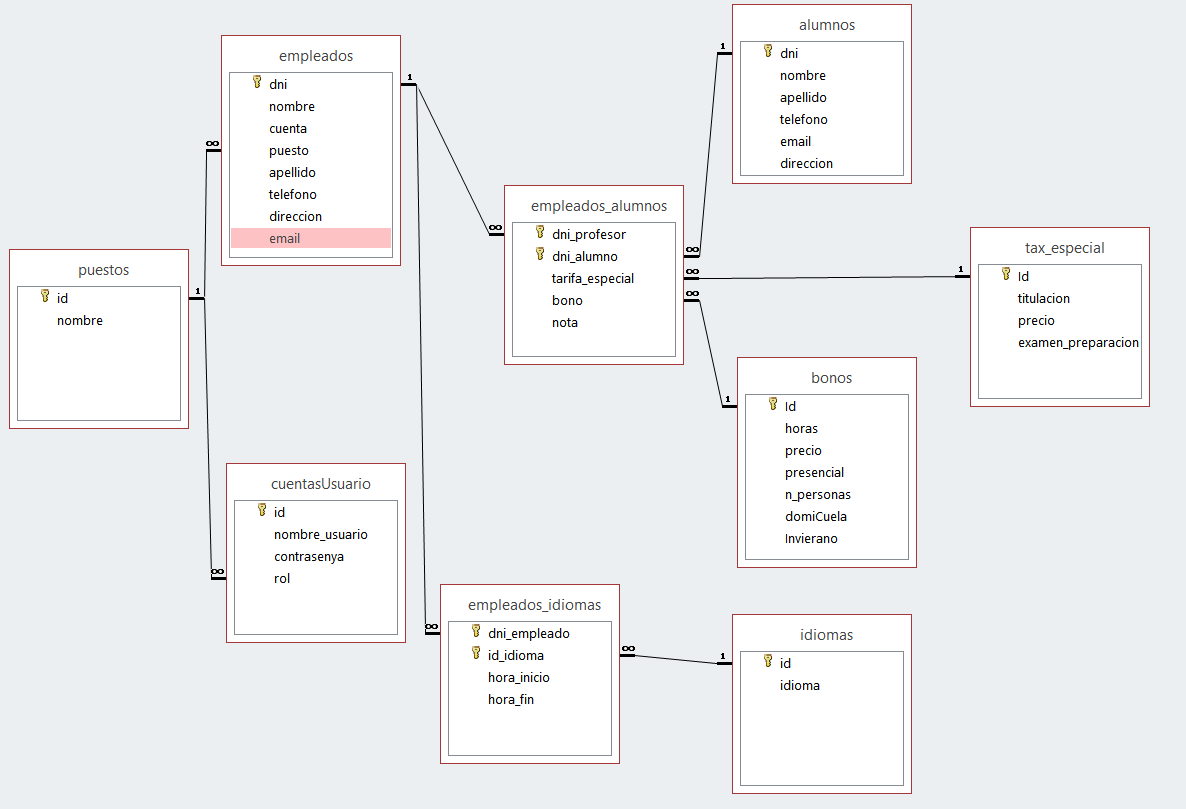




## ASI 6: Elaboración del modelo de datos

A continuación se muestra el modelo Entidad Relación de la base de datos a utilizar, en cual se pueden observar las tablas, sus campos y las relaciones que hay entre cada una de las entidades.

**Toda la información referente a los campos, tipos de datos, relaciones, dependencias, etc. Se encuentra en el documento llamado Informe de Base de Datos que viene adjunto a esta documentación.**



## ASI 8: Definición de interfaces de usuario

### ASI 8.3 Especificación de los formatos individuales de la interfaz de pantalla.

En la ventana inicial que se abre nada más iniciar la aplicación destacamos la presencia de un ProgressBar como se especificaba en el enunciado. Al cargar esta barra de progreso, la ventana se cierra y se abre otra, la pantalla de login. En esta pantalla de login tenemos un panel lateral que contiene la lista de usuarios que pueden iniciar sesión en la aplicación (tomados de la base de datos).

Al hacer click sobre un usuario veremos que se habilita el campo de texto contraseña así como el botón Acceder. Colocamos nuestra password de usuario y accedemos.

Una vez accedido se nos abrirá la ventana principal. Esta ventana puede mostrar 2 pestañas distintas: si tienes rol de jefe verás las pestañas de empleados así como la de alumnos. Si por el contrario tienes el rol de profesor sólo verás la pestaña de alumnos.

En la pestaña de empleados los jefes tendrán la posibilidad de ver un listado con los datos de los empleados registrados en la aplicación y tendremos las opciones de añadir, modificar y eliminar un empleado a la derecha.

En la pestaña de Alumnos un jefe tiene la posibilidad de ver todos los alumnos matriculados en el centro mientras que un profesor sólo puede ver sus alumnos. En el panel derecho tenemos de nuevo las opciones Añadir, modificar y eliminar un alumno y tenemos 2 opciones adicionales que son Asignar bono y Asignar tarifa especial.

Para más información sobre la funcionalidad en detalle de estas opciones, pulsar la tecla F1 en cualquier parte de la aplicación para abrir el CHM con la ayuda.

### ASI 8.4 Especificación de comportamiento dinámico de la interfaz.

El primer formulario (Form1) constará solamente de un ProgressBar controlado por un Timer. Cuando la progressbar llegue a 100 simplemente se cerrará esta ventana y se abrirá FormLogin.

FormLogin tiene una propiedad que es un objeto del tipo User que enviará a FormManagement al hacer click sobre el botón acceder..

FormManagement, la ventana principal de la aplicación, tendrá la misma propiedad, el objeto User recibido para saber quien ha iniciado sesión.

FormManagement le pasará a FormModify varios parámetros para decirle a esta pantalla qué funcionalidad debe realizar. El primero de estos parámetros será un Byte que definirá el modo en el que se ha abierto la ventana. Si la ventana se ha abierto mediante el botón Añadir, este Byte tomará el valor 0 . Si se ha abierto haciendo click sobre Modificar, será 1. Dependiendo del valor de este parámetro, la ventana se configurará de una manera o de otra.

El siguiente parámetro que recibe FormModify es un objeto. Este objeto representará al alumno o al empleado elegido de la lista en FormManagement.

Y el último parámetro será otro Byte que le indique a la ventana si el objeto recibido es un alumno o un empleado. Y como antes, dependiendo de su valor, la ventana se configurará de una manera o de otra.

FormModify antes de cerrarse abre la ventana donde se genera y se muestra el informe del alumno nuevo y también le dice a FormManagement que actualice las listas de empleados/alumnos.

Tanto FormBonos como FormSpecialTax sólo se podrán acceder desdela pestaña alumnos, por tanto, ambos formularios reciben respectivamente un objeto Alumno cada uno de ellos para saber a qué alumno de la base de datos se le va a asignar el bono/la tarifa especial que se elija en estas ventanas.

Ninguno de estos dos últimos formularios mencionados devuelve nada al cerrarse.

## ASI 10: Especificación del plan de pruebas.

# Diseño.

## DSI 1.6: Especificación del entorno tecnológico.

Para la codificación de la presente aplicación se ha utilizado Visual Studio 2017 como IDE. El lenguaje de codificación ha sido Visual Basic 6 y como motor de bases de datos se ha utilizado Microsoft Access 2013 y 2016.

## DSI 4: Diseño de clases.

### DSI 3.1 Identificación de clases asociadas a casos de uso.

El primer caso de uso presentado en el diagrama de arriba es el de gestión de clases del que salen a su vez Bono y Tarifa especial. Estas funcionalidades se han implementado en las clases Bonos y SpecialTax respectivamente. Estas clases se encuentran codificadas en la DLL anexa al proyecto principal. Adicionalmente las vistas que manejan esta funcionalidad son FormBonos para la funcionalidad de bonos y FormSpecialTex para la funcionalidad de tarifas especiales.

El siguiente caso de uso es el de la generación de informes. Por mera decisión de diseño no se ha codificado solamente una clase que implemente todos los métodos para la generación de informes sino que se han añadido a clases ya existentes. Los métodos que generan informes de alumnos están en la clase Alumnos de la DLL y el que implementa la generación del informe de profesores está en la clase CRUD\_employes. Los formularios que se encargan de mostrar los informes (y por tanto utilizar las funcionalidades codificadas mencionadas) tienen por nombre Form\_Crystal seguido de alumnos en el caso de los informes de los alumnos y empleados para el informe de empleados/profesores.

La funcionalidad Login asociada a su caso de uso se implementa en las clases Login y User especialmente diseñadas para tal fin. Y la funcionalidad logout en este caso no implica nada más que cerrar la ventana principal y volver a abrir la de login (FormLogin).

Para los dos últimos casos de uso (Gestión de alumnos y Gestión de empleados) se ha decidido implementar una sola ventana (FormManagement) para mostrar y proveer al usuario de las funcionalidades asociadas a sendos casos de uso.

En el caso de Gestión de alumnos, todas las funcionalidades (A/B/M, Asignar bono/tarifa especial) se implementan en la clase Alumnos y en las clases Bonos y SpecialTax como ya se ha mencionado, de la DLL. En el caso de Gestión de empleados, los métodos se han codificado en la clase CRUD\_empleados.

# Construcción

## CSI 2.1: Generación del código de los componentes.

### Resolución de requisitos y resolución de casos de uso.

A continuación se muestra la codificación de los diferentes requisitos descritos al principio del documento:

RF\_1:

Altas:



Modificaciones:



Bajas:



RF\_2:



RF\_3:

Asignar bono al alumno:



Asignar tarifa especial:



RF\_4: Este requisito, por la forma en la que se ha implementado la base de datos, se ha visto ya resuelto con lel código del RF\_3.

RF\_5:



RF\_6:



RF\_7:



C\_1: Esta característica es propia de la base de datos en sí.

C\_2:



C\_3:

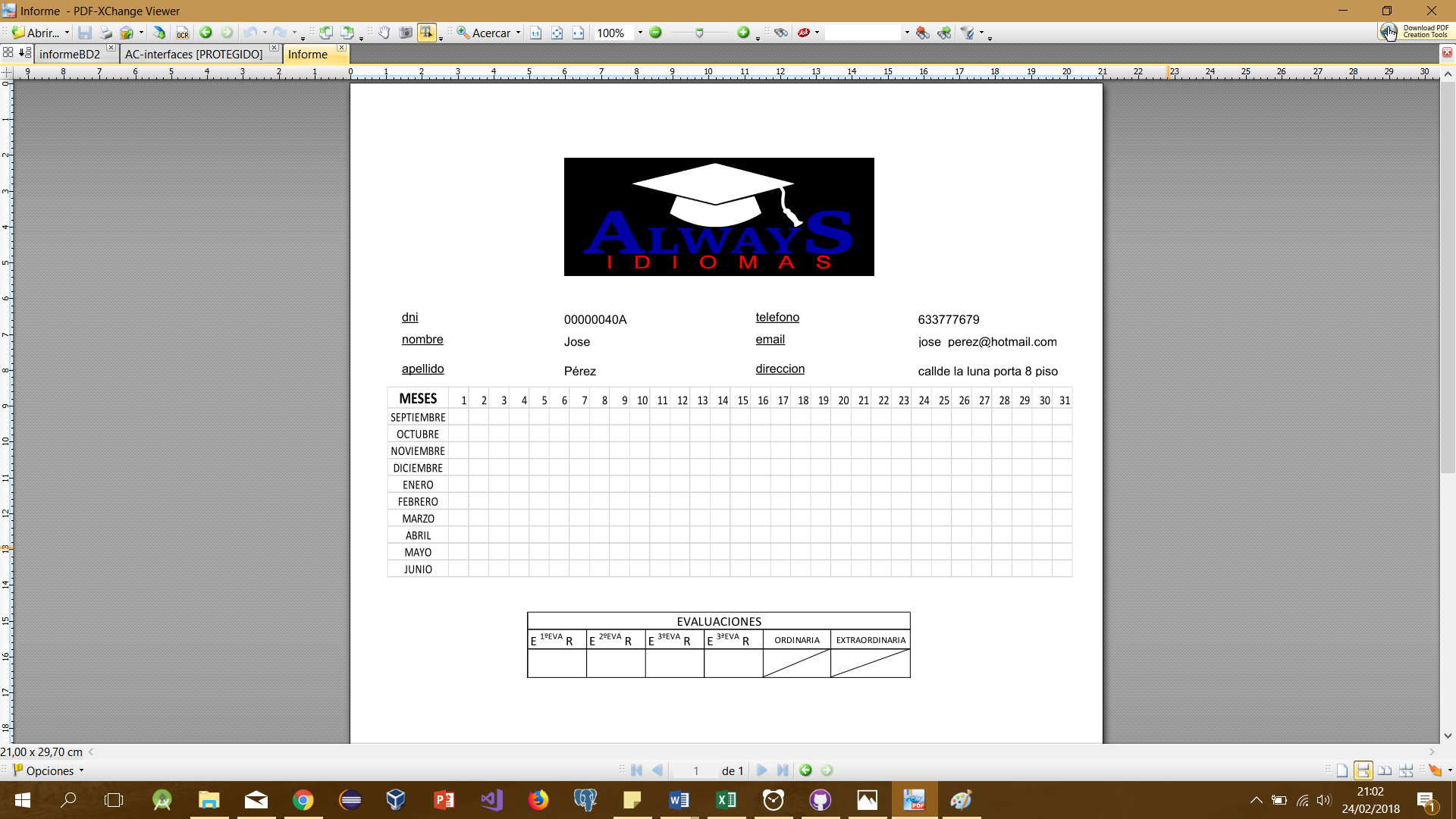


Las restricciones definidas se implementan a nivel de vista por tanto no hay un código específico asociado más allá de una sentencia IF.

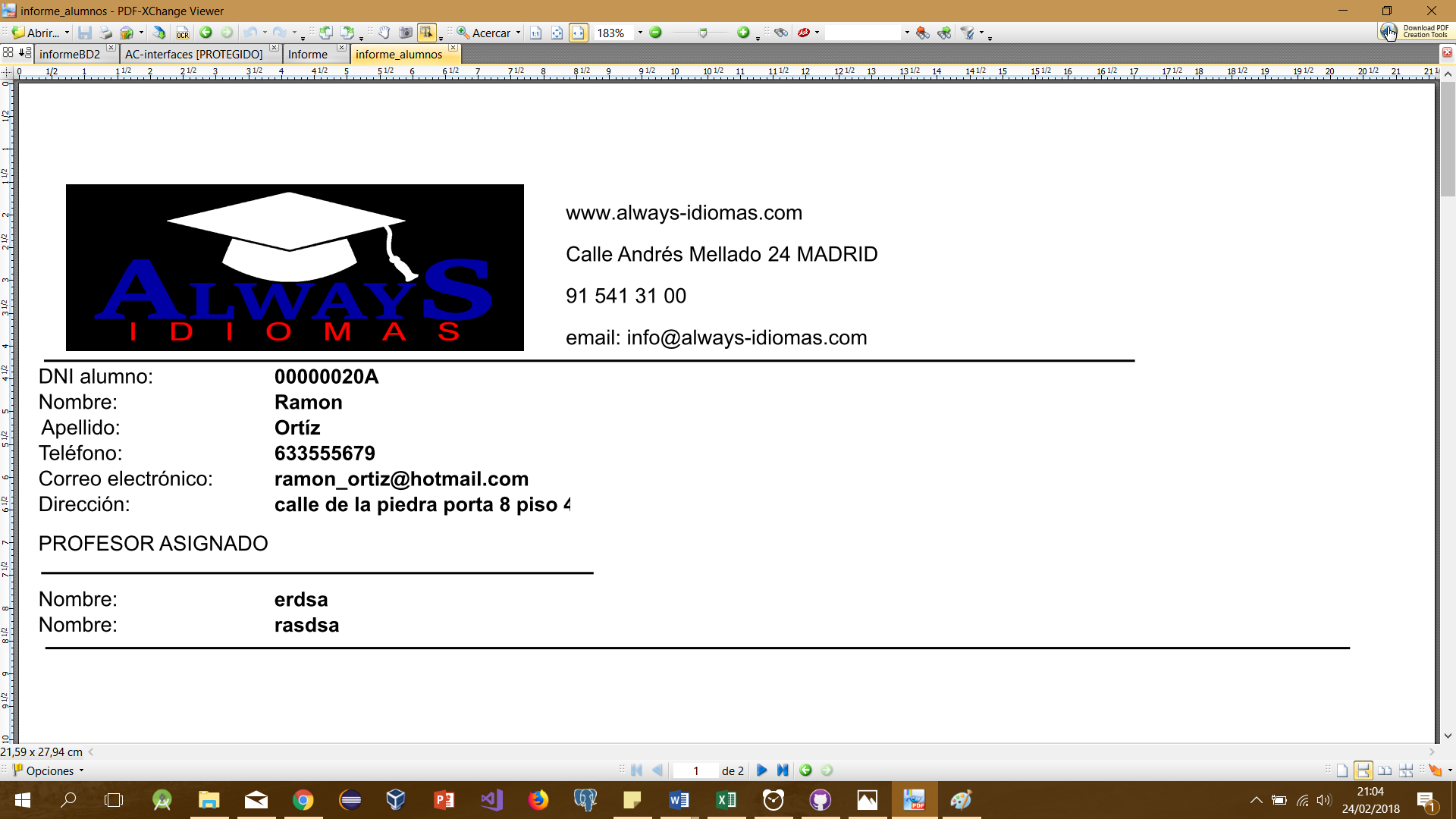
## CSI 6: Ejecución de pruebas.

## Elaboración de los manuales de usuario:

Este apartado se encuentra en el documento llamado Manual de usuario Always Academy adjunto a esta documentación.

INFORME GENREADO POR CADA ALUMNO NUEVO

INFORME DE TODOS LOS PROFESORES.



INFORME GENERADO DE TODOS LOS ALUMNOS



# Glosario.

1. Métrica V3: Metodología de Planificación, Desarrollo y Mantenimiento de sistemas de información.
2. LOPD: La Ley Orgánica 15/1999 de 13 de diciembre de Protección de Datos de Carácter Personal, (LOPD), es una ley orgánica española que tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales.
3. IEEE830: es una descripción completa del comportamiento del sistema a desarrollar .Incluye un conjunto de casos de uso que describen todas las interacciones que se prevén que los usuarios tendrán con el software.
4. Software: Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.
5. Base de datos: conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos. Cada base de datos se compone de una o más tablas que guarda un conjunto de datos.
6. CRUD: acrónimo de "Crear, Leer, Actualizar y Borrar" (del original en inglés: Create, Read, Update and Delete).
7. Validaciones formales: aquellas validaciones que independientemente del ámbito en el que se encuentre tiene que contener cierta estructura para que sea entendible, ejemplo: número de teléfono, DNI…
8. Validaciones de negocio: aquellas validaciones que dependen del ámbito en el que se encuentren, ejemplo: “los menores de 18 años no pueden comprar alcohol”, “El abono transporte cuesta 20 euros a aquellas personas menores de 26 años.”
9. Login: acceso individual a un sistema informático mediante la identificación del usuario utilizando credenciales provistas por el usuario.​
10. Logout: salida de un sistema informático mediante la identificación del usuario utilizando credenciales provistas por el usuario.​
11. UML: El lenguaje unificado de modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad
12. Bono: Papel, tarjeta o documento que se puede canjear por una cantidad de dinero, un objeto o un servicio, en este caso un servicio. El bono dispondrá de un número determinado de horas de clase (5, 10 ó 15) que el alumno podrá elegir a su gusto y adaptar otros parámetros a sus necesidades educativas.

# Web grafía.

1. VisualStudio: <https://www.visualstudio.com/es/>
2. LOPD: <https://www.boe.es/buscar/act.php?id=BOE-A-1999-23750>
3. Métrica V3: <https://goo.gl/cxRpgw>
4. UML: <http://www.uml.org/>
5. IEEE830: <https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>

1. A/B/M: son las iniciales de Alta, Baja, Modificación. [↑](#footnote-ref-1)
2. A/B/M: son las iniciales de Alta, Baja, Modificación. [↑](#footnote-ref-2)