**Proyecto 01: APIs de interoperabilidad para la Escuela de Baile**

Análisis de requerimientos

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Autor | Observaciones y/o agregados |
| 19 sep 2017 | 01 | Rafael Gamboa  rgamboa@itam.mx | Versión inicial de la especificación. |

Contenido

[1 Objetivo del proyecto: 1](#_Toc493529066)

[2 Requerimientos Funcionales: 1](#_Toc493529067)

[2.1 Alta de alumno: 1](#_Toc493529068)

[2.1.1 Happy path: 1](#_Toc493529069)

[2.1.2 Rutas alternas: 1](#_Toc493529070)

[2.1.3 Prototipo: 1](#_Toc493529071)

[2.2 Alta de profesor: 1](#_Toc493529072)

[2.2.1 Happy path: 1](#_Toc493529073)

[2.2.2 Rutas alternas: 1](#_Toc493529074)

[2.2.3 Prototipo: 1](#_Toc493529075)

[2.3 Alta de Baile: 1](#_Toc493529076)

[2.3.1 Happy path: 1](#_Toc493529077)

[2.3.2 Rutas alternas: 1](#_Toc493529078)

[2.3.3 Prototipo: 1](#_Toc493529079)

[2.4 Alta de curso: 2](#_Toc493529080)

[2.4.1 Happy path: 2](#_Toc493529081)

[2.4.2 Rutas alternas: 2](#_Toc493529082)

[2.4.3 Prototipo: 2](#_Toc493529083)

[2.5 Inscribir alumno a curso: 2](#_Toc493529084)

[2.5.1 Happy path: 2](#_Toc493529085)

[2.5.2 Rutas alternas: 2](#_Toc493529086)

[2.5.3 Prototipo: 2](#_Toc493529087)

[2.6 Asignación de profesor(es) a curso: 2](#_Toc493529088)

[2.6.1 Happy path: 2](#_Toc493529089)

[2.6.2 Rutas alternas: 2](#_Toc493529090)

[2.6.3 Prototipo: 2](#_Toc493529091)

[2.7 Pasar “lista de asistencia” a una sesión de un curso: 2](#_Toc493529092)

[2.7.1 Happy path: 2](#_Toc493529093)

[2.7.2 Rutas alternas: 2](#_Toc493529094)

[2.7.3 Prototipo: 2](#_Toc493529095)

[2.8 Acta de calificaciones del curso: 3](#_Toc493529096)

[2.8.1 Happy path: 3](#_Toc493529097)

[2.8.2 Rutas alternas: 3](#_Toc493529098)

[2.8.3 Prototipo: 3](#_Toc493529099)

[2.9 Pago de cuota para un curso por estudiante: 3](#_Toc493529100)

[2.9.1 Happy path: 3](#_Toc493529101)

[2.9.2 Rutas alternas: 3](#_Toc493529102)

[2.9.3 Prototipo: 3](#_Toc493529103)

[2.10 Kardex del estudiante: 3](#_Toc493529104)

[2.11 Generación de pago a profesor: 4](#_Toc493529105)

[2.11.1 Happy path: 4](#_Toc493529106)

[2.11.2 Rutas alternas: 4](#_Toc493529107)

[2.11.3 Prototipo: 4](#_Toc493529108)

[3 Requerimientos No Funcionales: 4](#_Toc493529109)

[4 Diseño preliminar: 4](#_Toc493529110)

[4.1 Estándares utilizados: 4](#_Toc493529111)

[5 Baterías de pruebas: 4](#_Toc493529112)

[5.1 Modalidad de la prueba, reactivos y resultados esperados. 4](#_Toc493529113)

# Objetivo del proyecto:

Contar con un conjunto de APIs para poder solicitar los servicios de la Escuela de Baile desde otra aplicación por medio de HTTP en la modalidad Web Service sobre <soap>.

# Requerimientos Funcionales:

## Alta de alumno:

### Happy path:

Da de alta un alumno a partir de su nombre, apellido paterno, apellido materno y nombre(s). El servicio debe asignar la clave al alumno y debe regresar esta clave como indicador de que todo resultó bien.

### Rutas alternas:

El alumno ya está dado de alta: En este caso la aplicación regresa la leyenda “Alumno ya existente”.

Prototipo:

(uso por POJO)

String ProxyEscBaile.altaAlumno(

String strNombre,

String strApPaterno,

String strApMaterno

)

## Alta de profesor:

### Happy path:

Da de alta un profesor a partir de su nombre, apellido paterno, apellido materno y nombre(s). El servicio debe asignar la clave al profesor y debe regresar esta clave como indicador de que todo resultó bien.

### Rutas alternas:

El profesor ya está dado de alta: En este caso la aplicación regresa la leyenda “Profesor ya existente”.

### Prototipo:

(uso por POJO)

String ProxyEscBaile.altaProfesor(

String strNombre,

String strApPaterno,

String strApMaterno

)

## Clave del alumno:

Se proporciona el apellido paterno, el apellido materno y el nombre del alumno y el servicio regresa la clave del alumno.

### Happy path:

### Rutas alternas:

### Prototipo:

## Clave de profesor:

Se proporciona el apellido paterno, el apellido materno y el nombre del profesor y el servicio regresa la clave del alumno.

### Happy path:

### Rutas alternas:

### Prototipo:

## Alta de Baile:

Se proporciona el nombre de un baile y su complejidad. El servicio responde con la clave del baile.

### Happy path:

### Rutas alternas:

### Prototipo:

## Alta de curso:

Se proporciona la clave del baile y el horario y cuántas sesiones del grupo a dar de alta. El servicio responde con los

### Happy path:

### Rutas alternas:

### Prototipo:

## Inscribir alumno a curso:

### Happy path:

### Rutas alternas:

### Prototipo:

## Asignación de profesor(es) a curso:

### Happy path:

### Rutas alternas:

### Prototipo:

## Pasar “lista de asistencia” a una sesión de un curso:

### Happy path:

### Rutas alternas:

### Prototipo:

## Acta de calificaciones del curso:

### Happy path:

### Rutas alternas:

### Prototipo:

## Pago de cuota para un curso por estudiante:

### Happy path:

### Rutas alternas:

### Prototipo:

## Kardex del estudiante:

## Generación de pago a profesor:

### Happy path:

### Rutas alternas:

### Prototipo:

# Requerimientos No Funcionales:

# Diseño preliminar:

## Estándares utilizados:

# Baterías de pruebas:

## Modalidad de la prueba, reactivos y resultados esperados.