ANEXO EXAMEN DE CERTIFICACIÓN

|  |  |
| --- | --- |
| Plan de estudio | Desarrollo de aplicaciones Fullstack JavaScript Trainee |
| Anexo | Caso Bootcamp Estudiantes Curso, día dos |

# Caso “Bootcamp Estudiantes Curso”

“Bootcamp Coders” es una academia dedicada a la entrega de servicios de capacitación orientados a la transformación digital, esto significa que se especializan principalmente en cursos de programación de sistemas web en diversos lenguajes de programación, como, por ejemplo, Fullstack Java, Fullstack Python, Fullstack JS, desarrollo Frontend con VUE, entre otros.

La academia tiene una existencia de aproximadamente 15 años en la industria de la capacitación, en la cual, aproximadamente hace 5 años se reestructuró para poder capacitar a personas bajo metodologías innovadoras denominadas bootcamp, activas y aula invertida.

Dada esta circunstancia, la demanda de capacitaciones creció considerablemente, gracias a esta actualización por requerimientos del mercado, por lo que, la academia se ve en la necesidad de crear un sistema web que les permita llevar el control de estudiantes, relatores, ayudantes y cursos, en dicho proyecto es en donde usted participará como desarrollador fullstack, junto a un equipo de profesionales con diversas competencias para poder implementar una solución apropiada según lo que esperan los gerentes de la organización.

El equipo se conformará por un profesional UX/UI, un desarrollador frontend, un líder de proyecto definido como un Scrum Master y usted que participará del proyecto como desarrollador fullstack.

El proyecto busca, como se mencionó previamente, generar un registro de estudiantes, cursos, relatores y ayudantes para poder generar un acceso rápido y persistente de los datos de los participantes de los diversos cursos y de los cursos propiamente tal. A continuación, se listan los requisitos funcionales de alto nivel:

* El sistema debe permitir consultar los cursos existentes, comuna y por consiguiente región asignada a dicho curso
* El sistema debe permitir la consulta de los tutores (ya sean ayudantes o relatores) que se encuentran participando de cada curso
* El sistema debe permitir identificar como se compone un curso, con respecto a los módulos que se impartirán dentro de este y los tutores que se harán cargo de cada uno

A la fecha, ya se ha avanzado en el proyecto y se cuenta con el siguiente avance:

* Ya se cuenta con un prototipo del aplicativo
* Existe un modelo de datos diseñado para dar solución a los requerimientos planteados hasta el momento
* Existe el script SQL que permite crear y poblar la base de datos, de momento, aun no se confirma con que motor de base de datos, se encuentran en alternativas viables MySQL y PostgreSQL, por lo que, momentáneamente se encuentran los script para ambos motores, por ahora, puede seleccionar el motor de base de datos que sea a su elección

# Modelo de Datos

A continuación, se presenta el modelo de datos diseñado por el arquitecto en conjunto con un analista

Diagrama

Descripción generada automáticamente

Como se puede observar, en primera instancia se definen los planes formativos que se ejecutan dentro de la academia, estos están compuestos por diversos módulos que le dan forma a cada plan

Desde cada plan formativo, se puede entender que se crean cursos basados en cada uno de los planes formativos, los cuales tienen una fecha de inicio y termino, en base al hecho que se crean desde un plan formativo, es que también se componen de módulos que le dan forma a cada curso y el total de hora que durará el curso

Para cada curso y módulo, se define un relator y ayudante, los cuales se relacionan entre si mediante la tabla “Curso\_Modulo\_Tutor”, en donde, mediante las claves primarias se genera una relación entre un curso, un módulo y los tutores (relator y ayudante)

En última instancia, cada curso se asocia a una comuna, lo que no limita a que los estudiantes puedan ser de otras comunas, dado que, los cursos son 100% remotos, de la misma forma, cada estudiante y tutor viven dentro de una comuna y por consiguiente en una región

# Requerimientos a desarrollar

El Scrum Master, que es el encargado de definir las tareas dentro del equipo, le ha solicitado a usted que realice las siguientes tareas

1. Realizar consultas a la base de datos
2. Crear monitor de estudiantes
3. Construir un algoritmo que permita calcular la cantidad de días que durará un curso acorde a una fecha de inicio y una fecha de termino
4. Crear API Rest que disponibilice la información del monitor de estudiantes

A continuación, se especifican con mayor detalle cada uno de los requerimientos:

1. Construcción de un algoritmo de cálculo de edad

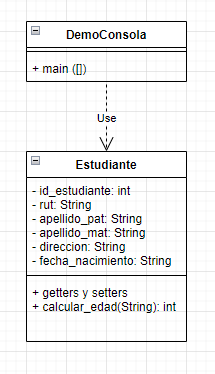
Una de las funcionalidades fundamentales del proyecto, es mantener la información de los estudiantes actualizada y disponible en el sistema web, de tal manera que cualquier usuario dentro de la organización pueda solicitar datos de estudiantes y contar con que el software le entregará lo solicitado de forma clara y rápida, para ello, se tiene en consideración incorporar más atributos a las diversas tablas definidas para complementar la información almacenada hasta el momento, como por ejemplo, la fecha de nacimiento de los estudiantes y/o tutores.

En base a lo expuesto, a usted se le solicita realizar un algoritmo que permita calcular la edad de los estudiantes desde su fecha de nacimiento a la fecha en la cual se solicita esta información.

Para realizar este cálculo se debe tener en consideración las siguientes instrucciones:

* La fecha de nacimiento se recibe como cadena de caracteres, se ha de manipular como un String para que de esta forma se pueda efectuar el cálculo de edad
* La fecha actual se ha de conseguir mediante instrucción del lenguaje a trabajar, esto significa que debemos investigar brevemente sobre alguna librería que nos pueda ayudar a obtener la fecha actual, en el caso que nos entregue algún tipo de dato diferente a una cadena de caracteres, se debe parsear para que se trabaje como un String
* Se debe obtener el año, mes y día de cada fecha para poder realizar los cálculos pertinentes, trabajar los tipos de datos apropiados para poder realizar las operaciones
* El método que calcule la edad debe retornar un entero con el resultado

Se le provee un diagrama de clases para poder trabajar con este algoritmo



Para hacer una demostración del algoritmo, cree una aplicación por consola que solicite 5 fechas por pantalla y se almacenen en objetos de tipo Estudiante, una fecha por un estudiante, para poder realizar el cálculo de la edad de cada uno, se debe mostrar la edad de cada estudiante por la consola

1. Construcción de API Rest para aplicaciones móviles

En vista y consideración que a futuro se piensa implementar una aplicación móvil que permita la consulta de los datos de un dispositivo móvil sin la necesidad de tener que ingresar al navegador, es que, se le solicita implementar una API Rest que entregue la misma información que el sistema de Estudiantes, recordar que debe recibir como parámetro la región y el curso

Para resolver este hito, tener en consideración trabajar con las 5 reglas necesarias para que el sistema sea calificable como API Rest Full, trabajar con el patrón de diseño apropiado para ello

Utilizar las herramientas necesarias para lograr la solución al requerimiento:

* NodeJS para trabajar con JS fuera del navegador
* NPM para la administración de paquetes, librerías
* ExpressJS para la construcción del backend bajo este marco de trabajo
* Librerías para la conexión a la base de datos, acorde a su selección, para MySQL o para PostgreSQL
* Definir apropiadamente los verbos HTTP para poder identificar el tipo de solicitud a realizarse a la API Rest