

Coding Challenge

- Vista



Bootstrap

Presentación de la aplicación frente al usuario. Consta de una presentación fácil de usar y amigable.

También las vistas

Coding Challenge

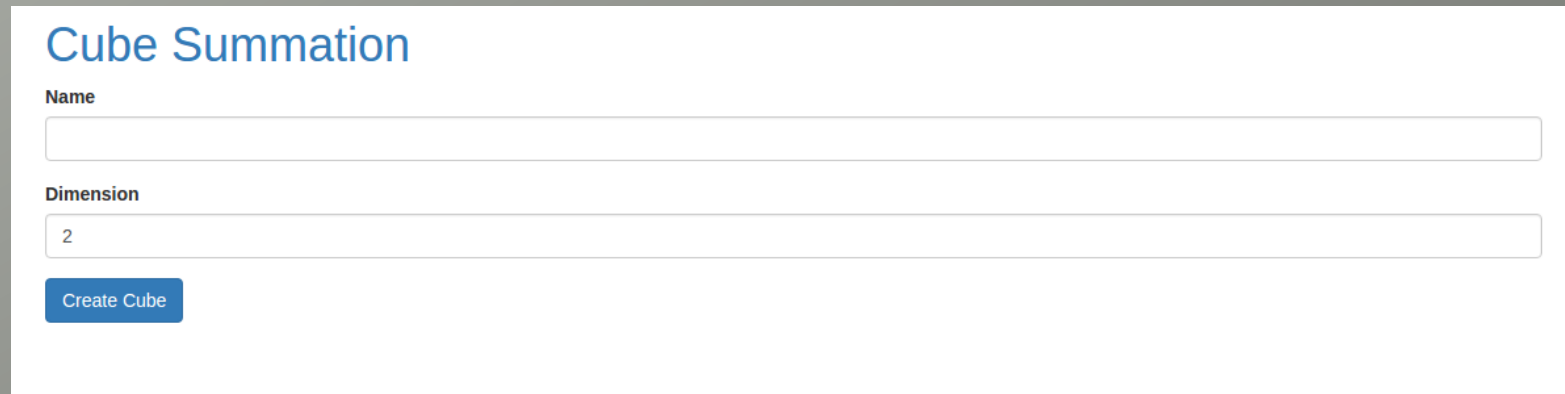
Lista de cubos

Cube Summation				
Create Cube				
Name	Dimension			
asd	2	Update	Query	Delete
Cubo 3	3	Update	Query	Delete
Otro cubo 4	4	Update	Query	Delete
Otro mas	4	Update	Query	Delete
Cubo 4	4	Update	Query	Delete

Listado de cubos creados, con enlaces para realizar las operaciones indicadas en el planteamiento del programa, también se dispone de la opción eliminar un determinado cubo.

Coding Challenge

Crear un cubo



Formulario de creación de un cubo. El formulario tiene un título "Cube Summation" en azul. Debajo del título, hay un campo de texto etiquetado "Name" que está vacío. Debajo de eso, hay un campo de texto etiquetado "Dimension" que contiene el número "2". Al final del formulario, hay un botón azul con el texto "Create Cube".

Formulario para la creación de un cubo, se le debe asignar un nombre (solo para mejorar su identificación) y su dimensión N (N^3).

El valor por defecto para todas las celdas es 0.

Coding Challenge

Update

Actualiza una posición del cubo, por un valor especificado.

Considerar que según lo planteado la posición inicial es (1,1,1).

Name: asd

Dimension: 2

Update

x

y

z

Value

Update

Coding Challenge

Query

asd

Dimension: 2

Query

X1

Y1

Z1

X2

Y2

Z2

Query

Permite realizar la suma desde un punto $A(x_1, y_1, z_1)$ a un punto $B(x_2, y_2, z_2)$ incluyendolos.

En ambas vistas (Query y Update) se puede visualizar el nombre del cubo y su dimension

Coding Challenge

- Capa de aplicación

Recibe las peticiones del usuario y posteriormente enviá las respuestas tras el proceso. Establece Todas las reglas y operaciones que se tendrán que cumplir

Coding Challenge

CubesController

Controlador de la aplicación. Maneja las solicitudes de cada una de las vistas y envío de información por medio de formularios

CubesController
Model: Cube View: /cubes
+Index() +Create() +Store(request) +update(id) +processUpdate(request,id) +query(id) +processQuery(id) +delete(id)

Coding Challenge

Relación Controlador-Vista

index()	GET /cubes/list Muestra el listado de los cubos creados
create()	GET /cubes/create Formulario, en el que se permite crear un cubo.
store(request)	POST /cubes/store Recibe una solicitud con la información del cubo a crear y procede a su almacenamiento en Base de datos.
update(id)	GET /cubes/update/{id} Recibe una solicitud con el id de un cubo en específico para actualizar el valor de una posición del cubo
processUpdate(request,id)	PUT /cubes/processUpdate/{id} Recibe el id del cubo y una solicitud con los valores a actualizar

Coding Challenge

query(id)	GET /cubes/query/{id} Muestra el formulario para el calculo de la sumatoria entre 2 puntos del cubo
processQuery(request,id)	POST /cubes/processQuery/{id} Toma los valores enviados desde el formulario ubicado en la ruta anterior, realiza la suma por medio de una clase
delete(id)	GET /cubes/delete/{id} Elimina un cubo especificado por el id

Coding Challenge

CubesController

CubeCreator

+Dimension

+Cube

+setDimension(dimension)

+setDefaultCube()

+getCube()

+update(x,y,z,val)

+getCubeSum(x1,y1,z1,x2,y2,z3)

+updateCube()

-to_php_array(val,inicio,final)

-to_pg_array(val)

Clase dedicada para la creación del cubo con su valor por defecto, actualización de celda, sumatoria de 2 puntos puntos, preparación del cubo para ser almacenado en la base de datos, y transformar el cubo cuando viene de la base datos a un arreglo tridimensional para poder ser operado,

Coding Challenge

CubesRequest

Establece las reglas (filtros) de los valores enviados desde el formulario *create*, antes de ser procesado por el controlador *CubesController*.

CubeUpdateRequest

Filtra la solicitud enviada por el formulario *update*.

CubeSumRequest

Filtra la solicitud enviada por el formulario *query*.

Coding Challenge

Capa de persistencia



Capa encargada del almacenamiento y recuperación de la información. Completamente independiente de la capa de aplicación.

Se ha creado una Base de datos con las tablas necesarias para el funcionamiento del sistemas.

Aunque existen varias tablas, la aplicación interactúa con una en específico llamada *Cubes*

Coding Challenge

Tabla *Cubes*

Cubes
Id (PK)
name
dimension
cube
created_at
updated_at

El campo id, otorga un identificador único a cada registro.

El campo *name* es para el nombre del cubo.

El campo cube almacena un arreglo de enteros (integer[]). Este tipo de dato es permitido en Postgres, por eso fue su elección.

Los campos created_at y updated_at son asignados automáticamente por laravel.