**PROYECTO PRUEBA**

**TECNOLOGÍAS WEB**

**Jefe de Experto:** Kenny Alejandro Ríos

**Actualización**

**2018 ©**

**Equipo Central Worldskills Colombia**

Hellman Yesid Rusinque Gamboa

Alejandra Gutiérrez Rengifo

Fredy García Reyes

Loren Jaramillo Bernal

**Contenido**

[**INTRODUCCIÓN** 3](#_Toc523222076)

[**2.** **FICHA DE DESCRIPCIÓN** 3](#_Toc523222077)

[**3. INSTRUCCIONES A LOS COMPETIDORES** 3](#_Toc523222078)

[**4. MODULOS DE COMPETENCIA** 4](#_Toc523222079)

[**4.1 Módulo E** 4](#_Toc523222080)

[**5. ESQUEMA DE MARCACION** 16](#_Toc523222081)

## **INTRODUCCIÓN**

En este proyecto prueba se describen los módulos del lado del servidor, los cuales cumplen con el descriptivo técnico internacional de la habilidad tecnologías web. En el documento encontrará las instrucciones, descripción, aclaraciones y esquema de calificación.

## **2.** **FICHA DE DESCRIPCIÓN**

El proyecto prueba posee 2 módulos, los cuales son desarrollados en un tiempo mínimo de 6 horas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Módulo** | **Día** | **Nombre** | **Horas** | **Porcentaje** |
| E | 3 | Lado del servidor - mañana | 3 | 12% |
| F | 3 | Lado del servidor - tarde | 3 | 10% |
| **TOTAL PROYECTO** | | | | **100%** |

**Porcentaje**

Está asociado a las secciones de especificación de estándares WSSS; sin embargo la construcción de los módulos puede contener la información de varias sesiones del WSSS

## **3. INSTRUCCIONES A LOS COMPETIDORES**

Para el desarrollo de este proyecto prueba, los competidores deben tener en cuenta el manual de salud, seguridad y ambiente, el reglamento interno de la habilidad y reglamento del aprendiz SENA.

## **4. MODULOS DE COMPETENCIA**

### **4.1 Módulo E**

|  |  |
| --- | --- |
| **Título del módulo** | Lado del servidor - mañana |
| **Tiempo máximo** | 3 horas |
| **Día de prueba** | Día 3 |
| **Porcentaje** | 12% |

**Objetivo:** Desarrollar una API Restful usando Framework de PHP.

**Descripción:**

El territorio colombiano tiene un encanto especial que atrapa fácilmente a extranjeros y nacionales. Cada región posee sus propios destinos turísticos que se han conservado a través del tiempo, manteniendo sus atractivos generación tras generación.

La marca país **COLOMBIA.CO** requiere el desarrollo de una plataforma web que le pueda ayudar a los turistas a obtener información general de los principales destinos turísticos de cada región del país.

El proyecto debe estar separado por una arquitectura del lado del servidor y una arquitectura del lado del cliente.

**Tecnologías permitidas:**

Framework:

* Laravel 5.5 o 5.6

Dependencias:

* Laravel cors
* Laravel Passport
* **Nota:** En el caso que deseen, pueden crear sus propios middleware para el proceso de autorización de consumos desde un cliente web y de igual manera para el proceso de autenticación con token. Las dependencias mencionadas anteriormente no son de uso obligatorio.

**Requisitos:**

La marca país **COLOMBIA.CO** entregará una lista de los servicios web que deben ser creados, al igual que cada una de las especificaciones requeridas: (entrada de datos, salida de datos, mensajes de OK, mensajes de error, entre otros).

En la plataforma deben existir tres tipos de roles de usuario:

* **VISITANTE** (usuario no registrado)
* **TURISTA** (usuario registrado)
* **ADMINISTRADOR** (usuario con todos los privilegios en el sistema)

Debe crear el usuario administrador, haciendo uso del seeder de laravel:

* nombre: administrador
* email: admin@colombia.co
* password: adminpass
* rol: administrador

Debe crear un usuario turista, haciendo uso del seeder de laravel:

* nombre: john doe
* email: johndoe@gmail.com
* password: johnpass
* rol: turista

Las especificaciones de los servicios web tienen las siguientes características:

* Backend totalmente independiente del frontend
* Información de cada servicio web:
  + URI del servicio web (endpoint)
  + Descripción
  + Método (verbo HTTP)
  + Parámetro en la URI (en el caso que aplique)
  + Headers
  + Request
  + Response
  + Status
  + Usuario
* Verbos HTTP:
  + GET
  + POST
  + PATCH
  + DELETE
* Ejemplos de Status:
  + **200: Respuesta de OK:** es utilizada para responder una operación exitosa.
  + **201: Respuesta de OK:** es utilizada para responder a un registro exitoso.
  + **400: Bad request:** indica que los parámetros enviados son inválidos.
  + **401: Unauthorized:** es retornado cuando no tenemos privilegios para consumir un recurso.
  + **422: Data not processed:** es retornado cuando la información no es procesada.
  + **500: Internal Server Error**: indica que se produjo un error inesperado en el servidor.

Si alguien ejecuta la url raíz de la aplicación (http://localhost:8000), debe aparecer el texto COLOMBIA.CO de manera centrada. (vista por defecto del framework)



**Lista de servicios web:**

**1. Registro de usuarios**

|  |  |
| --- | --- |
| **Número** | 1 |
| **Nombre del web service** | Registro de usuarios |
| **URI** | api/v1/registro |
| **Descripción** | El usuario visitante se registra en la plataforma y obtiene por defecto el rol de turista. |
| **Verbo HTTP** | POST |
| **Header** | Content-Type: application/json |
| **Usuarios** | VISITANTE |
| **Request body** | {  "nombre": "Pepito Pérez",  "email": "pepito@gmail.com",  "password": "123"  }  **Nota:**   * nombre: mínimo 3 caracteres * password: mínimo 3 caracteres * email: campo único (unique) |
| **Response** | **Correcto:**   * status: 201 * body:   {  "id": 1,  "nombre": "Pepito Pérez",  "email": "pepito@gmail.com",  "api\_token": "H1QLiV60SLwMu5ZCjmbC3UISxFOsyH5jFHQizgu7Eb"  }  **Incorrecto:** (Aplica para valores vacíos, tipos de datos incorrectos o un email ya registrado)   * status: 422 * body:   {  "mensaje": "información no procesada"  } |

**2. Autenticación:**

|  |  |
| --- | --- |
| **Número** | 2 |
| **Nombre del web service** | Autenticación de usuarios |
| **URI** | api/v1/auth/login |
| **Descripción** | El usuario registrado inicia sesión por medio de su email y su contraseña, obteniendo un token de validación para los servicios protegidos. |
| **Verbo HTTP** | POST |
| **Header** | Content-Type: application/json |
| **Usuarios** | TURISTA, ADMINISTRADOR |
| **Request body** | {  "email": "pepito@gmail.com",  "password": "1234"  } |
| **Response** | **Correcto:**   * status: 200 * body:   {  "id": 1,  "nombre": "administrador",  "email": "admin@colombia.co",  "api\_token": "H1QLiV60SLw5ZCjm6dI5bC3UISxFOsyH5jFHQizgu7Eb",  "rol": "ADMINISTRADOR"  }  **Incorrecto:**   * status: 400 * body:   {  "mensaje": "inicio de sesión invalido"  } |

|  |  |
| --- | --- |
| **Número** | 3 |
| **Nombre del web service** | Cierre de sesión |
| **URI** | api/v1/auth/logout*?token={TOKEN}* |
| **Descripción** | Se elimina la sesión del usuario autenticado y se destruye el token en la base de datos. |
| **Verbo HTTP** | GET |
| **Header** | Authorization Bearer Token |
| **Usuarios** | TURISTA, ADMINISTRADOR |
| **Request body** | N/A |
| **Response** | **Correcto:**   * status: 200 * body:   {  "mensaje": "cierre de sesión correcto"  }  **Incorrecto:**   * response status: 422 * body:   {  "mensaje": "información no procesada"  }  **No autorizado:** (Aplica para valores vacíos o incorrectos)   * status: 401 * body:   {  "mensaje": "usuario no autorizado"  } |

**3.     Departamentos:**

|  |  |
| --- | --- |
| **Número** | 4 |
| **Nombre del web service** | Crear un departamento |
| **URI** | api/v1/departamentos*?token={ TOKEN}* |
| **Descripción** | El administrador del sistema crea un nuevo departamento |
| **Verbo HTTP** | POST |
| **Header** | Authorization Bearer Token  Content-Type: application/json |
| **Usuarios** | ADMINISTRADOR |
| **Request body** | {  "departamento": "risaralda"  } |
| **Response** | **Correcto:**   * status: 201 * body:   {  "id": 1,  "departamento": "risaralda"  }  **Incorrecto:**   * status: 422 * body:   {  "mensaje": "información no procesada"  }  **No autorizado:**   * status: 401 * body:   {  "mensaje": "usuario no autorizado"  } |

|  |  |
| --- | --- |
| **Número** | 5 |
| **Nombre del web service** | Listar todos los departamentos |
| **URI** | api/v1/departamentos*?token={ TOKEN}* |
| **Descripción** | El administrador del sistema lista todos los departamentos registrados en la base de datos. |
| **Verbo HTTP** | GET |
| **Header** | Authorization Bearer Token |
| **Usuarios** | ADMINISTRADOR |
| **Request body** | N/A |
| **Response** | **Correcto:**   * status: 200 * body:   [  {  "id": 1,  "departamento": "caldas"  },  {  "id": 2,  "departamento": "risaralda"  },  …  ]  **Incorrecto:**   * status: 422 * body:   {  "mensaje": "información no procesada"  }  **No autorizado:**   * status: 401 * body:   {  "mensaje": "usuario no autorizado"  } |

|  |  |
| --- | --- |
| **Número** | 6 |
| **Nombre del web service** | Buscar un departamento |
| **URI** | api/v1/departamentos/{id}*?token={ TOKEN}* |
| **Descripción** | El administrador del sistema realiza la búsqueda de un departamento registrado en la base de datos. |
| **Verbo HTTP** | GET |
| **Header** | Authorization Bearer Token |
| **Usuarios** | ADMINISTRADOR |
| **Request body** | N/A |
| **Response** | **Correcto:**   * status: 200 * body:   {  "id": 1,  "departamento": "risaralda"  }  **Incorrecto:**   * status: 422 * body:   {  "mensaje": "información no procesada"  }  **No autorizado :**   * status: 401 * body:   {  "mensaje": "usuario no autorizado"  } |

|  |  |
| --- | --- |
| **Número** | 7 |
| **Nombre del web service** | Actualizar un departamento |
| **URI** | api/v1/departamentos/{id}*?token={TOKEN}* |
| **Descripción** | El administrador del sistema actualiza el registro de un departamento existente. |
| **Verbo HTTP** | PATCH |
| **Header** | Authorization Bearer Token  Content-Type: application/json |
| **Usuarios** | ADMINISTRADOR |
| **Request body** | {  "id": 1,  "departamento": "amazonas"  } |
| **Response** | **Correcto:**   * status: 200 * body:   {  "id": 1,  "departamento": "amazonas"  }  **Incorrecto:**   * status: 422 * body:   {  "mensaje": "información no procesada"  }  **No autorizado:**   * status: 401 * body:   {  "mensaje": "usuario no autorizado"  } |

**4.     Lugares:**

|  |  |
| --- | --- |
| **Número** | 8 |
| **Nombre del web service** | Crear un nuevo lugar |
| **URI** | api/v1/lugares*?token={ TOKEN}* |
| **Descripción** | El usuario crea una nuevo lugar, asociado al respectivo departamento. |
| **Verbo HTTP** | POST |
| **Header** | Authorization Bearer Token |
| **Usuarios** | TURISTA |
| **Request body**  **form-data** |  |
| **Response** | **Correcto:**   * status: 201 * body:   {  "id": 1,  "nombre": "santuario de las lajas",  "descripcion": "El Santuario de Nuestra Señora del Rosario de las Lajas …",  "imagen": "N9oZ8Xnb2GTgNjaGZAUfJLEL2FAS4GHlUrtslVPd.jpeg"  }  **Incorrecto:**   * status: 422 * body:   {  "mensaje": "información no procesada"  }  **No autorizado:**   * status: 401 * body:   {  "mensaje": "usuario no autorizado"  } |

|  |  |
| --- | --- |
| **Número** | 9 |
| **Nombre del web service** | Listar todos los lugares |
| **URI** | api/v1/lugares |
| **Descripción** | El usuario lista todos los lugares registrados en la base de datos. |
| **Verbo HTTP** | GET |
| **Header** | N/A |
| **Usuarios** | VISITANTE, TURISTA, ADMINISTRADOR |
| **Request body** | N/A |
| **Response** | **Correcto:**   * status: 200 * body:   [  {  "id": 1,  "nombre": "santuario de las lajas",  "descripcion": "El Santuario de Nuestra Señora del Rosario de las Laj",  "imagen": "N9oZ8Xnb2GTgNjaGZAUfJLEL2FAS4GHlUrtslVPd.jpeg"  }  {  "id": 2,  "nombre": "viaducto de pereira",  "descripcion": "lorem iopsum",  "imagen": "NHYS7644b2GTgNjaGZAUfJLEL2FAOHJSSAPd.jpeg"  }  ]  **Incorrecto:**   * status: 422 * body:   {  "mensaje": "información no procesada"  }  **No autorizado:**   * status: 401 * body:   {  "mensaje": "usuario no autorizado"  } |

|  |  |
| --- | --- |
| **Número** | 10 |
| **Nombre del web service** | Buscar un lugar |
| **URI** | api/v1/lugares/{id} |
| **Descripción** | El usuario consulta un lugar registrado en la base de datos. |
| **Verbo HTTP** | GET |
| **Header** | N/A |
| **Usuarios** | VISITANTE, TURISTA, ADMINISTRADOR |
| **Request body** | N/A |
| **Response** | **Correcto:**   * status: 200 * body:   {  "id": 1,  "nombre": "santuario de las lajas",  "descripcion": "El Santuario de Nuestra Señora del Rosario de las Laj",  "imagen": "N9oZ8Xnb2GTgNjaGZAUfJLEL2FAS4GHlUrtslVPd.jpeg"  }  **Incorrecto:**   * status: 422 * body:   {  "mensaje": "información no procesada"  }  **No autorizado:**   * status: 401 * body:   {  "mensaje": "usuario no autorizado"  } |

|  |  |
| --- | --- |
| **Número** | 11 |
| **Nombre del web service** | Eliminar un lugar |
| **URI** | api/v1/lugares/{id}*?token={TOKEN}* |
| **Descripción** | El administrador elimina un lugar existente. |
| **Verbo HTTP** | DELETE |
| **Header** | Authorization Bearer Token |
| **Usuarios** | ADMINISTRADOR |
| **Request body** | N/A |
| **Response** | **Correcto:**   * status: 200 * body:   {  "mensaje": "eliminación correcta"  }  **Incorrecto:**   * status: 422 * body:   {  "mensaje": "información no procesada"  }  **No autorizado:**   * status: 401 * body:   {  "mensaje": "usuario no autorizado"  } |

**5.     Usuarios:**

|  |  |
| --- | --- |
| **Número** | 12 |
| **Nombre del web service** | Listar todos los usuarios |
| **URI** | api/v1/usuarios/{id}*?token={TOKEN}* |
| **Descripción** | El administrador lista todos los turistas registrados en la base de datos. |
| **Verbo HTTP** | GET |
| **Header** | Authorization Bearer Token |
| **Usuarios** | ADMINISTRADOR |
| **Request body** | N/A |
| **Response** | **Correcto:**   * status: 200 * body:   [  {  "id": 1,  "nombre": "administrador",  "email": "admin@colombia.co",  "api\_token": "H1QLiV60SLw5ZCjm6dI5bC3UISxFOsyH5jFHQizgu7Eb",  "rol": "ADMINISTRADOR"  },  {  "id": 2,  "nombre": "John doe",  "email": "john@gmail.com",  "api\_token": "",  "rol": "TURISTA"  }  ]  **Incorrecto:**   * status: 422 * body:   {  "mensaje": "información no procesada"  }  **No autorizado:**   * status: 401 * body:   {  "mensaje": "usuario no autorizado"  } |

**Aclaraciones:**

* El nombre del proyecto debe ser “**XX\_ServerSide\_A**”, donde XX son las siglas de su regional. Este proyecto debe ser guardado en el escritorio del equipo, por lo cual la URL debe ser: http://<servername>/api/v1/<service>

Ejemplos:

* + **http://localhost/api/v1/users**
  + **http://localhost:8000/api/v1/users**
  + **http://127.0.0.1:8000/api/v1/users**
* La base de datos debe tener el nombre **“XX\_ServerSide\_A”,** donde XX son las siglas de su regional. Ejemplo: **AN\_ServerSide\_A**
* Emplear algoritmo de encriptación de contraseña bcrypt.
* Se debe garantizar la normalización de la base de datos. (por lo menos dos tablas relacionadas)
* Puede poblar la base de datos con la información suministrada en los medios, haciendo uso de factories, seeders o importaciones directamente a la base de datos.
* Las pruebas de los servicios del API Restful se deben realizar con POSTMAN. Debe usar el archivo **“Services\_ServerSide\_A.postman\_collection.json”,** teniendo en cuenta que con esta colección se revisarán los servicios web. El bearer token debe ser ingresado por usted en cada uno de los servicios que sea requerido.
* No importa el orden de los campos en las respuestas JSON.
* Todos los medios se encuentran ubicados en el escritorio, carpeta **ServerSide/Medios\_A**

**Medios suministrados:**

* **/departamentos/**
  + **departamentos.csv:** Información para poblar la base de datos.
* **diagrama.png:** Entidades y campos requeridos en la base de datos (usted debe realizar la normalización requerida)
* **/lugares/**
  + **prueba.docx:** Caso de ejemplo para probar el servicio web de crear un lugar.
  + **/bogotá/**
    - cerro-monserrate.jpg
    - bogotá.txt (información necesaria para crear el registro)
  + **/guaviare/**
    - tranquilandia.jpg
    - guaviare.txt (información necesaria para crear el registro)
  + **/huila/**
    - cascada-azufrada.jpg
    - huila.txt (información necesaria para crear el registro)
  + **/quindío/**
    - mirador-circasia.jpg
    - quindío.txt (información necesaria para crear el registro)
* **/fotos/**
  + fotografías de diferentes lugares para crear registros en el modo de desarrollo. Uso opcional
* **Services\_ServerSide\_A.postman\_collection.json**: (Colección de servicios web para consumir desde POSTMAN). En estos servicios no deben ser modificadas las url o lo campos, ya que sobre ellos se aplicarán los criterios de evaluación.

## **5. ESQUEMA DE MARCACION**

La evaluación de cada módulo se realizará de acuerdo a la siguiente tabla:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MÓDULO** | **CRITERIO** | **CALIFICACION POR MEDICION** | **CALIFICACION POR JUZGAMIENTO** | **TOTAL** |
| E1 | API General / Uso del Framework | 1,00 |  | 1,00 |
| E2 | Base de datos | 2,00 |  | 2,00 |
| E3 | Registro automático de usuarios | 0,5 |  | 0,5 |
| E4 | Servicios web | 7,80 |  | 7,80 |
| E5 | Seguridad y permisos | 0,70 |  | 0,70 |
|  |  |  |  | **12** |