

PLAN DE PRUEBAS

GRUPO 1 - José Antonio Llamas Álvarez, Miguel Lázaro,
Daniel Santano y José Luis Díez

Controlador	UnidadController
Funcionalidad a probar	mostrarUnidades()
Valores de prueba	username = "superAdmin", password = "1234"
Resultado esperado	Status 200.Ok
Miembro del grupo que implementará el test	José Antonio Llamas
Miembro del grupo que ha implementado el test	José Antonio Llamas

Controlador	UnidadController
Funcionalidad a probar	mostarUnidadPorCurso(@PathVariable Long id)
Valores de prueba	username = "superAdmin", password = "1234"
Resultado esperado	Status 200.Ok y contenido "application/json;charset=UTF-8"
Miembro del grupo que implementará el test	José Antonio Llamas
Miembro del grupo que ha implementado el test	José Antonio Llamas

Controlador	UnidadController
Funcionalidad a probar	nuevoUnidad(@RequestBody NuevaUnidadDto unidadDto)
Valores de prueba	username = "superAdmin", password = "1234"
Resultado esperado	Status 201.Created y comprobar que el usuario creado es igual que el esperado. (NO FUNCIONA PORQUE NO SE IMPLEMENTÓ EL SERVICIO)
Miembro del grupo que implementará el test	José Antonio Llamas
Miembro del grupo que ha implementado el test	José Antonio Llamas

Controlador	UnidadController
Funcionalidad a probar	editarUnidad(@PathVariable Long id, @Valid @RequestBody UnidadDto unidad)
Valores de prueba	username = "superAdmin", password = "1234" UnidadDto, Unidad
Resultado esperado	Cuando modificamos la Unidad, devuelva un Dto igual a unidadDto
Miembro del grupo que implementará el test	José Antonio Llamas
Miembro del grupo que ha implementado el test	José Antonio Llamas

Controlador	UnidadController
Funcionalidad a probar	deleteUnidad(@PathVariable Long id)
Valores de prueba	username = "superAdmin", password = "1234" Unidad
Resultado esperado	Cuando eliminamos la Unidad, devuelva status 204 (No content)
Miembro del grupo que implementará el test	José Antonio Llamas
Miembro del grupo que ha implementado el test	José Antonio Llamas

Controlador	CursoAcademicoController
Funcionalidad a probar	index()
Valores de prueba	username = "superAdmin", password = "1234"
Resultado esperado	status().isOk() y contentType("application/json;charset=UTF-8")
Miembro del grupo que implementará el test	José Antonio Llamas
Miembro del grupo que ha implementado el test	José Antonio Llamas

Controlador	CursoAcademicoController
Funcionalidad a probar	nuevoCursoAcademico(@RequestBody CursoAcademicoDto cursoAcademicoDto)
Valores de prueba	username = "superAdmin", password = "1234" CursoAcademico, CursoAcademicoDto
Resultado esperado	status().201 y CursoAcademico equals a CursoAcademicoDto transformado
Miembro del grupo que implementará el test	José Antonio Llamas
Miembro del grupo que ha implementado el test	José Antonio Llamas

Controlador	CursoAcademicoController
Funcionalidad a probar	delete(@PathVariable Long id)
Valores de prueba	username = "superAdmin", password = "1234" CursoAcademico
Resultado esperado	status().204
Miembro del grupo que implementará el test	José Antonio Llamas
Miembro del grupo que ha implementado el test	José Antonio Llamas

Controlador	RegistrarseController
Funcionalidad a probar	registrarse(@RequestBody CreateUserEntityDto createUserEntityDto)
Valores de prueba	username = "superAdmin", password = "1234" CreateUserDto, User, UserDto
Resultado esperado	Comprobar que el resultado de registrar el CreateUserDto es igual a User status().201
Miembro del grupo que implementará el test	José Antonio Llamas
Miembro del grupo que ha implementado el test	José Antonio Llamas

Controlador	SuperAdminController
Funcionalidad a probar	crearNuevoAdmin(@RequestBody CreateUserEntityDto createUserEntityDto) ,
Valores de prueba	username = "superAdmin", password = "1234" createUserDto, User, UserDto
Resultado esperado	Comprobar que el resultado de registrar el CreateUserDto es igual a User status().201
Miembro del grupo que implementará el test	José Antonio Llamas
Miembro del grupo que ha implementado el test	José Antonio Llamas

Controlador	Etapas
Funcionalidad a probar	mostrarEtapas()
Valores de prueba	username = "superAdmin", password = "1234,EtapasDto,Etapas
Resultado esperado	Status 200.Ok
Miembro del grupo que implementará el test	José Luis Díez
Miembro del grupo que ha implementado el test	José Luis Díez

Controlador	Etapas
Funcionalidad a probar	mostarEtapasCursoAcademico(@PathVariable Long id)
Valores de prueba	username = "superAdmin", password = "1234,EtapasDto,Etapas
Resultado esperado	Status 200.Ok
Miembro del grupo que implementará el test	José Luis Díez
Miembro del grupo que ha implementado el test	José Luis Díez

Controlador	Etapa
Funcionalidad a probar	nuevoEtapa(@RequestBody NuevaEtapaDto etapaDto)
Valores de prueba	username = "superAdmin", password = "1234,EtapaDto,Etapa
Resultado esperado	Status 201.Created y comprobar que la etapa creada es igual que el resultado esperado.
Miembro del grupo que implementará el test	José Luis Díez
Miembro del grupo que ha implementado el test	José Luis Díez

Controlador	Etapa
Funcionalidad a probar	deleteEtapa(@PathVariable Long id)
Valores de prueba	username = "superAdmin", password = "1234,,Etapa
Resultado esperado	status()204.isNoContent()
Miembro del grupo que implementará el test	José Luis Díez
Miembro del grupo que ha implementado el test	José Luis Díez

Controlador	Etapa
Funcionalidad a probar	mostrarEtapas()
Valores de prueba	username = "superAdmin", password = "1234,null
Resultado esperado	status().404.isNoContent()
Miembro del grupo que implementará el test	José Luis Díez
Miembro del grupo que ha implementado el test	José Luis Díez

Controlador	Procesos
Funcionalidad a probar	mostrarProcesos(@PathVariable Long id)
Valores de prueba	username = "superAdmin", password = "1234,Proceso,ProcesoDto
Resultado esperado	status().isOk()
Miembro del grupo que implementará el test	José Luis Díez
Miembro del grupo que ha implementado el test	José Luis Díez

Controlador	Procesos
Funcionalidad a probar	deleteEtapa(@PathVariable Long id)
Valores de prueba	username = "superAdmin", password = "1234,Proceso
Resultado esperado	status().isNoContent()
Miembro del grupo que implementará el test	José Luis Díez
Miembro del grupo que ha implementado el test	José Luis Díez

Controlador	Procesos
Funcionalidad a probar	crearProceso(@RequestBody ProcesoDto procesoDto)
Valores de prueba	username = "superAdmin", password = "1234,Proceso,ProcesoDto,CursoAcademicoDto,CursoAcademico
Resultado esperado	Status 201.Created y comprobar que el proceso creado es igual que el resultado esperado.
Miembro del grupo que implementará el test	José Luis Díez
Miembro del grupo que ha implementado el test	José Luis Díez

Controlador	UserEntity
Funcionalidad a probar	getOneUser(@PathVariable Long id)
Valores de prueba	username = "User", password = "1234",UserEntity,UserEntityDto
Resultado esperado	status().isOk()
Miembro del grupo que implementará el test	José Luis Díez
Miembro del grupo que ha implementado el test	José Luis Díez

Controlador	UserEntity
Funcionalidad a probar	getMyUser(@PathVariable String username)
Valores de prueba	username = "User", password = "1234",UserEntity,UserEntityDto,Optional
Resultado esperado	status().isOk()
Miembro del grupo que implementará el test	José Luis Díez
Miembro del grupo que ha implementado el test	José Luis Díez

Controlador	ColegioController
Funcionalidad a probar	obtenerColegios()
Valores de prueba	username = "usuario", password = "1234", roles = "USER"
Resultado esperado	status().isOk() 200
Miembro del grupo que implementará el test	Daniel Santano Fernández
Miembro del grupo que ha implementado el test	Daniel Santano Fernández

Controlador	ColegioController
Funcionalidad a probar	obtenerColegios()
Valores de prueba	username = "usuario", password = "1234", roles = "USER"
Resultado esperado	status().isNoContent() 204
Miembro del grupo que implementará el test	Daniel Santano Fernández
Miembro del grupo que ha implementado el test	Daniel Santano Fernández

Controlador	AdminController
Funcionalidad a probar	getAllUsuarios()
Valores de prueba	username = "usuario", password = "1234", roles = "USER"
Resultado esperado	status().isOk() 200
Miembro del grupo que implementará el test	Daniel Santano Fernández
Miembro del grupo que ha implementado el test	Daniel Santano Fernández

Controlador	AdminController
Funcionalidad a probar	getAllUsuarios()
Valores de prueba	username = "usuario", password = "1234", roles = "USER"
Resultado esperado	status().isNoContent() 204
Miembro del grupo que implementará el test	Daniel Santano Fernández
Miembro del grupo que ha implementado el test	Daniel Santano Fernández

Controlador	PsmController
Funcionalidad a probar	crearPsm()
Valores de prueba	username = "admin", password = "1234", roles = "ADMIN", PsmDto dto
Resultado esperado	status().isCreatedt() 201
Miembro del grupo que implementará el test	Daniel Santano Fernández
Miembro del grupo que ha implementado el test	Daniel Santano Fernández

Controlador	PsmController
Funcionalidad a probar	crearPsm()
Valores de prueba	username = "admin", password = "1234", roles = "ADMIN", PsmDto dto
Resultado esperado	status().isNoContent() 204
Miembro del grupo que implementará el test	Daniel Santano Fernández
Miembro del grupo que ha implementado el test	Daniel Santano Fernández

Controlador	PsmController
Funcionalidad a probar	obtenerPsmByCursoAcademicoldAndEvaluacion()
Valores de prueba	username = "admin", password = "1234", roles = "ADMIN", ObtenerPsmDtoByCaldAndEv dto
Resultado esperado	status().isOk() 200
Miembro del grupo que implementará el test	Daniel Santano Fernández
Miembro del grupo que ha implementado el test	Daniel Santano Fernández

Controlador	PsmController
Funcionalidad a probar	obtenerPsmByCursoAcademicoldAndEvaluacion()
Valores de prueba	username = "admin", password = "1234", roles = "ADMIN", ObtenerPsmDtoByCaldAndEv dto
Resultado esperado	status().isNoContent() 204
Miembro del grupo que implementará el test	Daniel Santano Fernández
Miembro del grupo que ha implementado el test	Daniel Santano Fernández

Controlador	PsmController
Funcionalidad a probar	index()
Valores de prueba	username = "admin", password = "1234", roles = "ADMIN"
Resultado esperado	status().isOk() 200
Miembro del grupo que implementará el test	Daniel Santano Fernández
Miembro del grupo que ha implementado el test	Daniel Santano Fernández

Controlador	PsmController
Funcionalidad a probar	index()
Valores de prueba	username = "admin", password = "1234", roles = "ADMIN"
Resultado esperado	status().isNoContent() 204
Miembro del grupo que implementará el test	Daniel Santano Fernández
Miembro del grupo que ha implementado el test	Daniel Santano Fernández

Controlador	PsmController
Funcionalidad a probar	delete()
Valores de prueba	username = "admin", password = "1234", roles = "ADMIN", Long id
Resultado esperado	status().isNoContent() 204
Miembro del grupo que implementará el test	Daniel Santano Fernández
Miembro del grupo que ha implementado el test	Daniel Santano Fernández

Controlador	PsmController
Funcionalidad a probar	delete()
Valores de prueba	username = "admin", password = "1234", roles = "ADMIN", Long id
Resultado esperado	status().isBadRequest() 400
Miembro del grupo que implementará el test	Daniel Santano Fernández
Miembro del grupo que ha implementado el test	Daniel Santano Fernández

Controlador	CursoController
Funcionalidad a probar	Mostrar cursos por id de curso académico
Valores de prueba	CursoDto cdto = CursoDto.builder().id(3L).nombreCurso("2º DAM").nombreEtapa(" ").peso(1).unidades(unidadesDto).build();
Resultado esperado	200
Miembro del grupo que implementará el test	Miguel Lázaro Domínguez
Miembro del grupo que ha	Miguel Lázaro Domínguez

implementado el test	
-----------------------------	--

Controlador	CursoController
Funcionalidad a probar	borrar Cursos
Valores de prueba	Curso c = Curso.builder().id(1L).nombreCurso("2º DAM").peso(1).unidades(null).build();
Resultado esperado	204
Miembro del grupo que implementará el test	Miguel Lázaro Domínguez
Miembro del grupo que ha implementado el test	Miguel Lázaro Domínguez

Controlador	CursoController
Funcionalidad a probar	Crear Curso
Valores de prueba	CursoDto preSaved = CursoDto.builder().id(1L).nombreCurso("2º DAM").peso(1).unidades(null).build()
Resultado esperado	201
Miembro del grupo que implementará el test	Miguel Lázaro Domínguez
Miembro del grupo que ha implementado el test	Miguel Lázaro Domínguez

Controlador	PuntoControlControler
Funcionalidad a probar	Mostrar puntos de control por id de curso academico
Valores de prueba	PuntoControlDto pcdto = PuntoControlDto.builder().id(1L).fecha(null).colegioDto(null).psm(null).build();
Resultado esperado	200
Miembro del grupo que implementará el test	Miguel Lázaro Domínguez
Miembro del grupo que ha implementado el test	Miguel Lázaro Domínguez

Controlador	PuntoControlControler
Funcionalidad a probar	Mostrar puntos de control vacios
Valores de prueba	PuntoControlDto pcdto = PuntoControlDto.builder().id(1L).fecha(null).colegioDto(null).psm(null).build();
Resultado esperado	200
Miembro del grupo que implementará el test	Miguel Lázaro Domínguez
Miembro del grupo que ha implementado el test	Miguel Lázaro Domínguez

Controlador	PuntoControlControler
--------------------	-----------------------

Funcionalidad a probar	Mostrar puntos de control vacios (no hay datos)
Valores de prueba	EmptyList
Resultado esperado	204
Miembro del grupo que implementará el test	Miguel Lázaro Domínguez
Miembro del grupo que ha implementado el test	Miguel Lázaro Domínguez

Controlador	PuntoControlControler
Funcionalidad a probar	Borrar punto de control
Valores de prueba	PuntoControlDto pcdto = PuntoControlDto.builder().id(1L).fecha(null).colegioDto(null).psm(null).build();
Resultado esperado	200
Miembro del grupo que implementará el test	Miguel Lázaro Domínguez
Miembro del grupo que ha implementado el test	Miguel Lázaro Domínguez

Controlador	PuntoControlControler
Funcionalidad a probar	Crear punto de control
Valores de prueba	PuntoControlDto pcdto = PuntoControlDto.builder().id(1L).fecha(null).colegioDto(null).psm(null).build();
Resultado esperado	201
Miembro del grupo que implementará el test	Miguel Lázaro Domínguez
Miembro del grupo que ha implementado el test	Miguel Lázaro Domínguez

Controlador	IndicadorController
Funcionalidad a probar	Mostrar indicadores
Valores de prueba	IndicadorDto idto = IndicadorDto.builder().id(1L).nombreIndicador("dafs").esPorcentaje(true).proceso(null).build();
Resultado esperado	200
Miembro del grupo que implementará el test	Miguel Lázaro Domínguez
Miembro del grupo que ha implementado el test	Miguel Lázaro Domínguez

Controlador	IndicadorController
Funcionalidad a probar	Mostrar indicadores (sin datos)
Valores de prueba	EmptyList
Resultado esperado	204
Miembro del grupo que implementará el test	Miguel Lázaro Domínguez
Miembro del grupo que ha implementado el test	Miguel Lázaro Domínguez

Controlador	IndicadorController
Funcionalidad a probar	Mostrar indicadores por proceso Id
Valores de prueba	IndicadorDto idto = IndicadorDto.builder().id(1L).nombreIndicador("dafs").esPorcentaje(true).proceso(null).build();
Resultado esperado	200
Miembro del grupo que implementará el test	Miguel Lázaro Domínguez
Miembro del grupo que ha implementado el test	Miguel Lázaro Domínguez

Controlador	IndicadorController
Funcionalidad a probar	Borrar por id
Valores de prueba	IndicadorDto idto = IndicadorDto.builder().id(1L).nombreIndicador("dafs").esPorcentaje(true).proceso(null).build();
Resultado esperado	204
Miembro del grupo que implementará el test	Miguel Lázaro Domínguez
Miembro del grupo que ha implementado el test	Miguel Lázaro Domínguez

Controlador	IndicadorController
Funcionalidad a probar	Editar indicador
Valores de prueba	IndicadorDto idto = IndicadorDto.builder().id(1L).nombreIndicador("dafs").esPorcentaje(true).proceso(null).build();
Resultado esperado	200
Miembro del grupo que implementará el test	Miguel Lázaro Domínguez
Miembro del grupo que ha implementado el test	Miguel Lázaro Domínguez

Controlador	IndicadorController
Funcionalidad a probar	Nuevo indicador
Valores de prueba	IndicadorDto idto = IndicadorDto.builder().id(1L).nombreIndicador("dafs").esPorcentaje(true).proceso(null).build();
Resultado esperado	201
Miembro del grupo que implementará el test	Miguel Lázaro Domínguez
Miembro del grupo que ha implementado el test	Miguel Lázaro Domínguez

Controlador	IndicadorController
Funcionalidad a probar	Nuevo indicadores (Vacio)
Valores de prueba	IndicadorDto idto = IndicadorDto.builder().id(1L).nombreIndicador("dafs").esPorcentaje(true).proceso(null).build();
Resultado esperado	204
Miembro del grupo que implementará el test	Miguel Lázaro Domínguez
Miembro del grupo que ha implementado el test	Miguel Lázaro Domínguez