Mi Página | Cursos | Universidades e instituciones | Conócenos | Soporte Desarrollo de servicios en la nube con HTML5, Javascript y node.js Inicio Syllabus Foro Blog Módulos Modulo 4. Introducción a HTTP y a los Servidores Web; Introducción a express y al Middleware Static; Introducción a REST; Aplicaciones express.js y Composición de Middlewares; Formularios GET y POST; Parámetros Ocultos Modulo 0. Introducción al curso, al programa y al Sistema Operativo UNIX Prueba realizada Modulo 1. Introducción a JavaScript de servidor y a node.js. Sentencias, Tu resultado en el test ha sido: 100% Variables, Booleanos, Números, Strings y Funciones Has superado el test. Modulo 2. Introducción a JavaScript de Tus respuestas servidor y a node.js. Bucles, Clases predefinidas, Objetos, Propiedades y Supongamos que en nuestro ordenador local arrancamos la siguiente aplicación express: Métodos; Prototipos y Clases; Arrays; JSON; Funciones como Objetos y Cierres (Closures) var express = require('express'); Modulo 3. Modulos node.js; var app = express(); Expresiones Regulares; Eventos, app.get('/:id1(\\d+)/:id2?', function (req,res){ Entorno de Ejecución y Concurrencia res.send(req.params.id1 + (req.params.id2 | | "")); en node.js; Ficheros y Flujos app.get('*', function (req, res){res.send('Nadie');}); Modulo 4. Introducción a HTTP y a los app.listen(80); Servidores Web; Introducción a express y al Middleware Static; Introducción a REST; Aplicaciones Con qué texto responderá esta aplicación a la siguiente URL: http://localhost/4caminos express.js y Composición de Middlewares; Formularios GET y POST; 9"3P2" **Parámetros Ocultos** "3" Tema 0: Transparencias del módulo "P2" Tema 1. introducción a HTTP Tema 1. Cuestionario obligatorio "37signals" Tema 2. Servidor Web "4caminos" Tema 2. Cuestionario obligatorio "cuatrocaminos" ✓ Tema 3. Introducción a express.js y al "Nadie" middelware static Tema 3. Cuestionario obligatorio Tema 4. Introducción a REST Supongamos que en nuestro ordenador local arrancamos la siguiente aplicación express: Tema 4. Cuestionario obligatorio Tema 5. Aplicaciones REST con express.js var express = require('express'); var app = express(); Tema 5. Cuestionario obligatorio app.get('/:id1(\\d+)/:id2?', function (req,res){ Tema 6. Acceso a campos de la ruta res.send(req.params.id1 + (req.params.id2 | | "")); Tema 6. Cuestionario obligatorio app.get('*', function (req, res){res.send('Nadie');}); Tema 7. Composición y ejecución de app.listen(80); middlewares Tema 8. Formulario GET Con qué texto responderá esta aplicación a la siguiente URL: http://localhost/3P2 Tema 9. URL encode Tema 10. Formulario POST "3P2" Tema 11. Parámetro oculto y method override "3" Ejercicio P2P Obligatorio "P2" "37signals" Modulo 5. Gestión de versiones de proyectos con git y GITHUB; Proyecto, "4caminos" Espacio de Trabajo y Versiones "cuatrocaminos" (Commit); Arboles y Ramas de un proyecto; Repositorios Remoto y "Nadie" colaboración a través de GITHUB Ø Modulo 6. Proyecto Quiz I: Patrón Modelo-Vista-Controlador (MVC); Supongamos que en nuestro ordenador local arrancamos la siguiente aplicación express: generación del proyecto con expressgenerator; Primera Página y Primera Pregunta; Despliegue en la nube var express = require('express'); (Heroku) var app = express(); app.get('/:id1(\\d+)/:id2?', function (req,res){ Modulo 7. Proyecto Quiz II: La Base de res.send(req.params.id1 + (req.params.id2 | | "")); Datos (DB), Tablas, sequelize.js y SQLite; Despliegue en Heroku app.get('*', function (req, res){res.send('Nadie');}); utilizando Postgres; Presentación de app.listen(80); Listas de Quizes y Autoload Modulo 8. Proyecto Quiz III: Gestión de Listas de Quizes, Creación, Edición y Con qué texto responderá esta aplicación a la siguiente URL: http://localhost/37signals Borrado 9"3P2" Modulo 9. Proyecto Quiz IV: Creación y Moderación de Comentarios a Quizes; **"3"** Relaciones entre Tablas de la Base de "P2" Datos; Sesiones, Autenticación y Autorización; HTTP Seguro (HTTPS) "37signals" "4caminos" "cuatrocaminos" "Nadie" Supongamos que en nuestro ordenador local arrancamos la siguiente aplicación express: var express = require('express'); var app = express(); app.get('/:id1(\\d+)/:id2?', function (req,res){ res.send(req.params.id1 + (req.params.id2 | | "")); app.get('*', function (req, res){res.send('Nadie');}); app.listen(80); Con qué texto responderá esta aplicación a la siguiente URL: http://localhost/signals "3P2" **"3"** "37signals" "4caminos" "cuatrocaminos" "Nadie" Supongamos que en nuestro ordenador local arrancamos la siguiente aplicación express: var express = require('express'); var app = express(); app.get('/:id1(\\d+)/:id2?', function (req,res){ res.send(req.params.id1 + (req.params.id2 | | """)); app.get('*', function (req, res){res.send('Nadie');}); app.listen(80); Con qué texto responderá esta aplicación a la siguiente URL: http://localhost/3/P2 "3P2" **"3"** "P2" "37signals" "4caminos" "cuatrocaminos" "Nadie" Supongamos que en nuestro ordenador local arrancamos la siguiente aplicación express: var express = require('express'); var app = express(); app.get('/:id1(\\d+)/:id2?', function (req,res){ res.send(req.params.id1 + (req.params.id2 | | """)); app.get('*', function (req, res){res.send('Nadie');}); app.listen(80); Con qué texto responderá esta aplicación a la siguiente URL: http://localhost/37/signals "3P2" **"3"** "P2" "37signals" "4caminos" "cuatrocaminos" "Nadie"

Supongamos que en nuestro ordenador local arrancamos la siguiente aplicación express:

var express = require('express'); var app = express(); app.get('/:id1(\\d+)/:id2?', function (req,res){ res.send(req.params.id1 + (req.params.id2 | | """)); app.get('*', function (req, res){res.send('Nadie');}); app.listen(80); Con qué texto responderá esta aplicación a la siguiente URL: http://localhost/P2 "3P2" **"3"** "P2" "37signals" "4caminos" "cuatrocaminos" "Nadie"

res.send(req.params.id1 + (req.params.id2 | | """)); app.get('*', function (req, res){res.send('Nadie');}); app.listen(80);

Supongamos que en nuestro ordenador local arrancamos la siguiente aplicación express:

var express = require('express');

var express = require('express');

app.get('/:id1(\\d+)/:id2?', function (req,res){

var app = express();

"Nadie"

var app = express();

app.get('/:id1(\\d+)/:id2?', function (req,res){

var app = express();

Con qué texto responderá esta aplicación a la siguiente URL: http://localhost/4/caminos "3P2" **"3"** "P2" "37signals" "4caminos" "cuatrocaminos" "Nadie"

res.send(req.params.id1 + (req.params.id2 | | "")); app.get('*', function (req, res){res.send('Nadie');}); app.listen(80);

Supongamos que en nuestro ordenador local arrancamos la siguiente aplicación express:

Con qué texto responderá esta aplicación a la siguiente URL:http://localhost/cuatro/caminos "3P2" **"3"** "P2" "37signals" "4caminos" "cuatrocaminos"

Supongamos que en nuestro ordenador local arrancamos la siguiente aplicación express: var express = require('express');

app.get('*', function (req, res){res.send('Nadie');}); app.listen(80); Con qué texto responderá esta aplicación a la siguiente URL: http://localhost/3 "3P2"

app.get('/:id1(\\d+)/:id2?', function (req,res){

res.send(req.params.id1 + (req.params.id2 | | ""));

"3"

"P2" "37signals" "4caminos" "cuatrocaminos" "Nadie" anterior Siguiente 😜

Telefonica

Telefónica Educación Digital

2012-2015 Miríada X Aviso legal Política de cookies Política de privacidad

№ Santander

uni>ersia