



“Estados HTTP”

:: UNIDAD 3: Fundamentos de programación de dispositivos móviles

:: ACTIVIDAD COMPLEMENTARIA 2

MATERIA: :: PROGRAMACIÓN DE DISPOSITIVOS MÓVILES ::

CLAVE: 1668

LICENCIATURA EN INFORMÁTICA

PLAN 2012

REALIZO: Emmanuel Alejandro Pérez Hernández

No. 423142118

Grupo: 8691

ASESOR: MARTINEZ FERNANDEZ JUAN MANUEL

domingo, 2 de noviembre de 2025

UNIDAD 3. Actividad Complementaria 2

a) Investiga los estados HTTP comunes.

Los códigos de estado HTTP se clasifican en 5 categorías principales.

Código	Categoría	Significado
200	Éxito (2xx)	OK. La solicitud fue exitosa y el servidor devuelve el contenido solicitado.
301	Redirección (3xx)	Movido Permanentemente. El recurso ha sido trasladado a una nueva URL de forma permanente.
302	Redirección (3xx)	Encontrado (Redirección Temporal). El recurso se encuentra temporalmente en otra URL.
304	Redirección (3xx)	No Modificado. El contenido no ha cambiado desde la última vez que se accedió (se usa la versión en caché).
400	Error Cliente (4xx)	Solicitud Incorrecta. El servidor no puede entender la solicitud debido a un error del cliente (ej: sintaxis inválida).
401	Error Cliente (4xx)	No Autorizado. Se requiere autenticación (login) para acceder al recurso.
403	Error Cliente (4xx)	Prohibido. El servidor entiende la solicitud, pero se niega a autorizarla (no tiene permisos).
404	Error Cliente (4xx)	No Encontrado. El recurso solicitado no existe en el servidor.
500	Error Servidor (5xx)	Error Interno del Servidor. El servidor encontró una condición inesperada que le impide completar la solicitud.
503	Error Servidor (5xx)	Servicio No Disponible. El servidor no puede manejar la solicitud en este momento (por mantenimiento o sobrecarga).

b) ¿Qué significa un estado 201?

El código de estado 201 Created (Creado) pertenece a la categoría de Éxito (2xx).

- Significado: Indica que la solicitud del cliente (por ejemplo, enviar un formulario o hacer una petición **POST** a una API) se ha procesado con éxito y ha resultado en la creación de uno o más nuevos recursos en el servidor.
- Cuándo ocurre: Es la respuesta típica y esperada después de una acción que crea algo nuevo, como:



- Publicar un nuevo artículo en un blog.
- Registrar un nuevo usuario en un sitio web.
- Subir un archivo a un servidor.
- Agregar un producto a una base de datos a través de una API.
- Qué debe incluir la respuesta: Idealmente, la respuesta del servidor debe incluir un encabezado **Location** que contenga la URL del nuevo recurso creado, para que el cliente sepa exactamente dónde encontrarlo.

c) Ejecuta la app Node.js con CodeSpace y apunta la dirección del túnel.

The screenshot shows the VS Code interface with the following details:

- Explorador:** Shows the project structure for "MI-APLICACION-EXPRESS" with files like node_modules, .gitignore, base.sqlite3, index.js, package-lock.json, and package.json.
- Editor:** Displays the content of index.js, which includes code to start an Express server on port 3000 and log the message "Aplicación corriendo en http://localhost:\$port".
- Terminal:** Shows the command `@Getzemax-98 → /workspaces/mi-aplicacion-express (main) \$ npm start` being run, followed by the output: "actividad-10@1.0.0 start" and "node index.js". It also shows the application is running on port 3000.
- Output:** Shows the application is connected to a SQLite database and has created a table named "Tareas".
- Puertos:** Shows a tunnel is active on port 3000.

Servidor Node.js y Túnel

The screenshot shows the VS Code interface with the following details:

- Explorador:** Shows the project structure for "MI-APLICACION-EXPRESS" with files like node_modules, .gitignore, base.sqlite3, index.js, package-lock.json, and package.json.
- Editor:** Displays the content of index.js, which includes code to start an Express server on port 3000 and log the message "Aplicación corriendo en http://localhost:\$port".
- Terminal:** Shows the command `@Getzemax-98 → /workspaces/mi-aplicacion-express (main) \$ ngrok http 3000` being run, followed by the output: "ERROR: https://shiny-turkeys-join.loca.lt" and "Enfocar carpeta en el explorador (ctrl + clic)".
- Output:** Shows the command `@Getzemax-98 → /workspaces/mi-aplicacion-express (main) \$ npm install -g localtunnel` being run, followed by the output: "added 22 packages in 901ms", "3 packages are looking for funding", and "run 'npm fund' for details".
- Puertos:** Shows a tunnel is active on port 3000 with the URL "http://shiny-turkeys-join.loca.lt".

Túnel activo



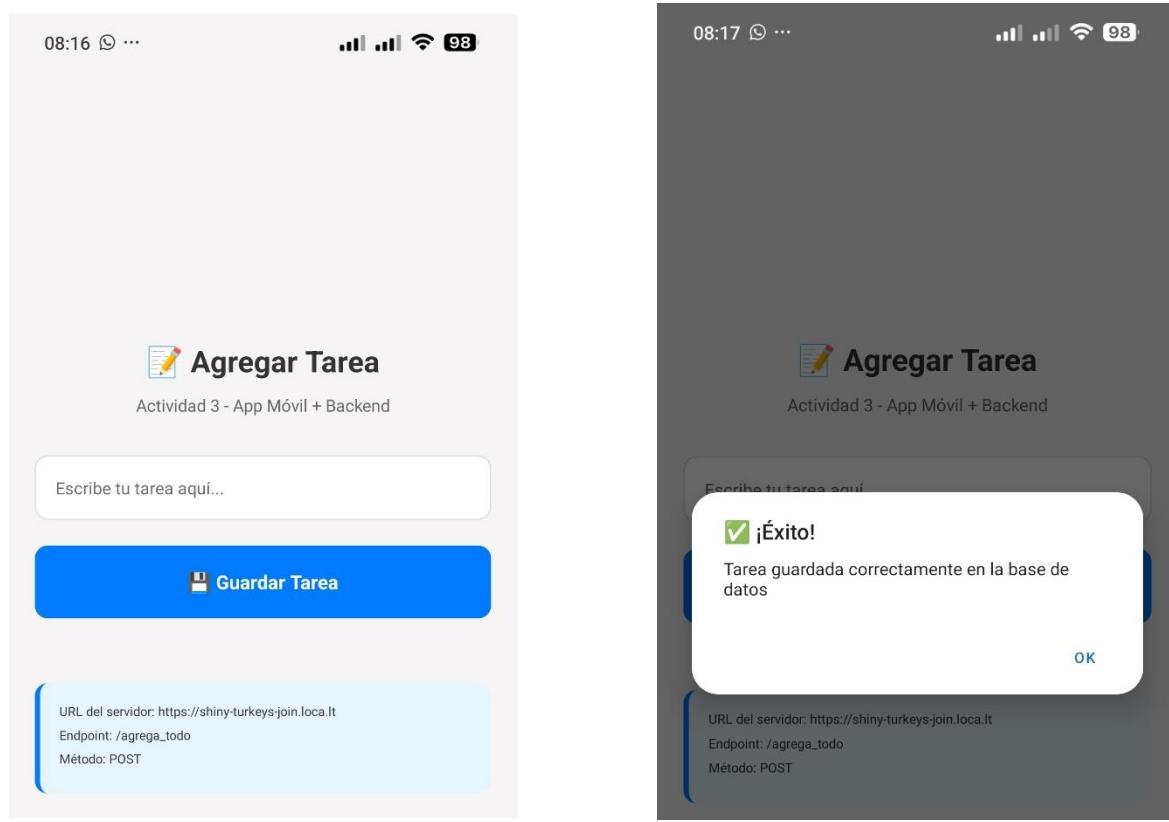
d) Modifica la app móvil con la dirección del túnel y verifica que los datos se guarden.

```

Códigos de estado HTTP comunes e | Getzemax-98/mi-aplicacion-express | index.js - mi-aplicacion-express | delicious green soygent - Snack
vigilant-space-couscous-7vj694jx9qv93p7pr.github.dev
Importantes Redes Sociales Entretenimiento Bancos OtakuZone Videojuegos Omar Lic. Info Compilador IA MCOC Adobe Pokemon
Explorador Bienvenido JS index.js M x
MI-APLICACION-EXPRESS [CODESPACES: VIGILANT ...]
node_modules .gitignore base.sqlite3
index.js package-lock.json package.json
ESQUEMA LÍNEA DE TIEMPO
index.js ...
120 );
121
122 //Corremos el servidor en el puerto 3000
123 const port = 3000;
124
125 app.listen(port, () => {
126   console.log('Aplicación corriendo en http://localhost:' + port)
127 })
128
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS 1
@Getzemax-98 →/workspaces/mi-aplicacion-express (main) $ cat package
@Getzemax-98 →/workspaces/mi-aplicacion-express (main) $ npm start
> actividadad-10@0.0.0 start
> node index.js
Aplicación corriendo en http://localhost:3000
Conectado a la base de datos SQLite.
Tabla tareas creada o ya existente.
Tarea recibida desde app móvil: Tarea Unidad 3 Actividad Complementaria 2
Tarea agregada exitosamente. ID: 1
Lín. 128, col. 1 Espacios: 4 UTF-8 LF JavaScript Finalizar configuración Diseño: Latin American
n... node
+ ... x
Lin. 128, col. 1 Espacios: 4 UTF-8 LF JavaScript Finalizar configuración Diseño: Latin American
8:17 p.m. 08:17 p.m.

```

App funcionando



App en Android

App en Android 2



Código de la app

```
import React, { useState } from 'react';
import { View, Text, TextInput, TouchableOpacity, Alert, StyleSheet } from 'react-native';

// ⚠ IMPORTANTE: Cambia esta URL por tu túnel de LocalTunnel
const API_URL = 'https://shiny-turkeys-join.loca.lt';

export default function App() {
  const [tarea, setTarea] = useState("");
  const [cargando, setCargando] = useState(false);

  const agregarTodo = async () => {
    if (tarea.trim() === "") {
      Alert.alert('✖ Error', 'Por favor escribe una tarea');
      return;
    }

    setCargando(true);

    try {
      console.log('Enviando tarea:', tarea);

      const response = await fetch(`${API_URL}/agrega_todo`, {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify({
          tarea: tarea
        }),
      });

      console.log('Estado de respuesta:', response.status);
      console.log('URL usada:', `${API_URL}/agrega_todo`);

      // ACTIVIDAD E: Verificar si el estado es 201
      if (response.status === 201) {
        const data = await response.json();
        console.log('Respuesta del servidor:', data);

        Alert.alert(
          '✅ ¡Éxito!',
          'Tarea guardada correctamente en la base de datos',
          [{ text: 'OK', onPress: () => console.log('Alerta cerrada') }]
        )
      }
    } catch (error) {
      console.error(error);
    }
  }
}
```



```

    );
    setTarea(""); // Limpiar el campo
} else {
    Alert.alert('✖ Error', `Error del servidor: ${response.status}`);
}

} catch (error) {
    console.error('Error de conexión:', error);
    Alert.alert(
        '∅ Error de conexión',
        'No se pudo conectar al servidor. Verifica:\n1. Tu conexión a internet\n2. Que el servidor esté corriendo\n3. Que la URL del túnel sea correcta'
    );
} finally {
    setCargando(false);
}
};

return (
    <View style={styles.container}>
        <Text style={styles.title}>✍ Agregar Tarea</Text>
        <Text style={styles.subtitle}>Actividad 3 - App Móvil + Backend</Text>

        <TextInput
            style={styles.input}
            placeholder="Escribe tu tarea aquí..."
            value={tarea}
            onChangeText={setTarea}
            editable={!cargando}
        />

        <TouchableOpacity
            style={[styles.boton, cargando && styles.botonDeshabilitado]}
            onPress={agregarTodo}
            disabled={cargando}
        >
            <Text style={styles.botonTexto}>
                {cargando ? '⌚ Guardando...' : '💾 Guardar Tarea'}
            </Text>
        </TouchableOpacity>

        <View style={styles.info}>
            <Text style={styles.infoText}>URL del servidor: {API_URL}</Text>
            <Text style={styles.infoText}>Endpoint: /agrega_todo</Text>
            <Text style={styles.infoText}>Método: POST</Text>
        </View>

```



```

        </View>
    );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 20,
    justifyContent: 'center',
    backgroundColor: '#f5f5f5',
  },
  title: {
    fontSize: 28,
    fontWeight: 'bold',
    marginBottom: 10,
    textAlign: 'center',
    color: '#333',
  },
  subtitle: {
    fontSize: 16,
    marginBottom: 30,
    textAlign: 'center',
    color: '#666',
  },
  input: {
    borderWidth: 1,
    borderColor: '#ddd',
    padding: 15,
    marginBottom: 20,
    borderRadius: 10,
    backgroundColor: 'white',
    fontSize: 16,
  },
  boton: {
    backgroundColor: '#007AFF',
    padding: 18,
    borderRadius: 10,
    alignItems: 'center',
    marginBottom: 30,
  },
  botonDeshabilitado: {
    backgroundColor: '#ccc',
  },
  botonTexto: {
    color: 'white',
    fontSize: 18,
    fontWeight: 'bold',
  }
});

```



```

},
info: {
  marginTop: 20,
  padding: 15,
  backgroundColor: '#e8f4fd',
  borderRadius: 10,
  borderLeftWidth: 4,
  borderLeftColor: '#007AFF',
},
infoText: {
  fontSize: 12,
  color: '#333',
  marginBottom: 5,
},
});

```

e) Muestra una alerta de éxito si el estado de la respuesta es 201.

```

// ACTIVIDAD E: Verificar si el estado es 201
if (response.status === 201) {
  const data = await response.json();
  console.log('Respuesta del servidor:', data);

  Alert.alert(
    '✅ ¡éxito!',
    'Tarea guardada correctamente en la base de datos',
    [{ text: 'OK', onPress: () => console.log('Alerta cerrada') }]
  );
  setTarea('');
} else {
  Alert.alert('✖ Error', `Error del servidor: ${response.status}`);
}

} catch (error) {
  console.error('Error de conexión:', error);
  Alert.alert(
    '⚠️ Error de conexión',
    'No se pudo conectar al servidor. Verifica:\n1. Tu conexión a internet\n2. Que el servidor esté corriendo\n3. Que la URL del túnel sea correcta'
  );
} finally {
  setCargando(false);
}
};

```

Alerta éxito 201

f) Enlace de tu Snack en un PDF

<https://snack.expo.dev/@getzemax98/modificar-app-movil>



I. REFERENCIAS BIBLIOGRÁFICAS

- Schmitz, M. (2020, May 26). *Los 10 códigos de estado HTTP más comunes*. Dotcom-Monitor Web Performance Blog. [https://www.dotcom-monitor.com/blog/es/los-10-most-common-http-status-codes/#:~:text=Los%20c%C3%B3digos%20de%20estado%20HTTP%20\(Hypertext%20Transfer,una%20solicitud%20debido%20a%20un%20problema%20inesperado](https://www.dotcom-monitor.com/blog/es/los-10-most-common-http-status-codes/#:~:text=Los%20c%C3%B3digos%20de%20estado%20HTTP%20(Hypertext%20Transfer,una%20solicitud%20debido%20a%20un%20problema%20inesperado)
- Duncan, C. (2023, December 12). *13 Common HTTP Status Codes + Explanations*. Seomator.com; SEOmator.



or%20cumpliera%20la%20solicitud.&text=No%20implementado:%20el%20servidor%20no,para%20cumplir%20con%20la%20solicitud.

Hitesh B. (2025, June 5). *Status Code 201 vs 200: Key Differences Explained*. ResellerClub Blog. <https://www.resellerclub.com/blog/what-is-the-difference-between-http-201-vs-200/#:~:text=%C2%BFQu%C3%A9son%20los%20c%C3%B3digos%20de,su%20funci%C3%B3n%20con%20prop%C3%B3sitos%20diferentes>.

