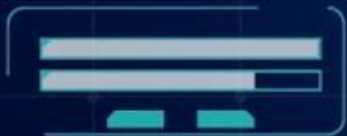


Curso de Desarrollo Móvil en Android





Bienvenido alumno

Sesiones



Completadas



5

Exámenes

4

Tareas

4



José María Cruz Parada
chema.cruz.p@gmail.com



Resumen de curso



Reglamento



Temario



Sesiones

0



Exámenes

4

Tareas

4



Sesión 1



Fundamentos de Java

Sesión 3



Ciclo de vida del Activity

Sesión 5



Fragments

Sesión 2



Fundamentos de Android

Sesión 4



Componentes gráficos

Sesión 6



Servicios Web

Sesión 7

Sesión 8



REST

Retrofit



¿Qué es un Rest?





REST

Retrofit



Operaciones basadas en recursos





REST

Retrofit



Arquitectura cliente-servidor





REST

Retrofit



Operaciones basadas en recursos





REST

Retrofit



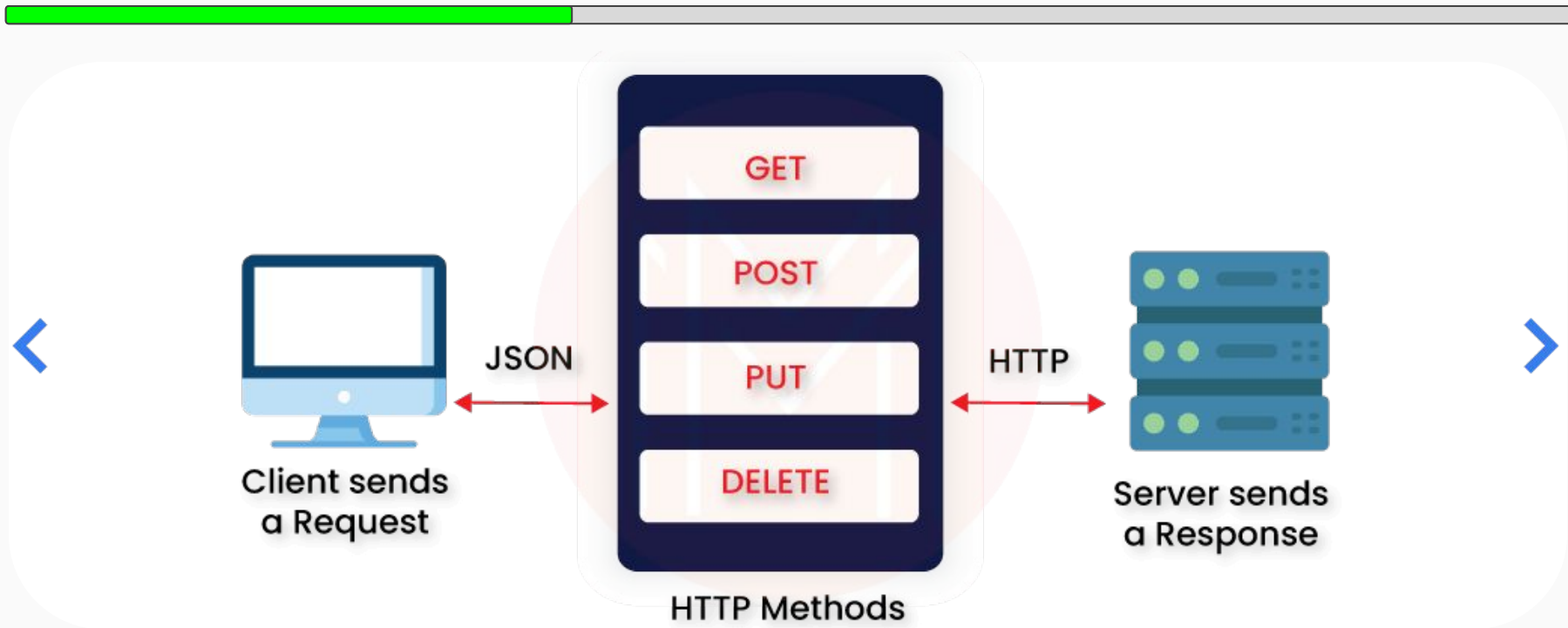
Operaciones basadas en recursos





REST

Retrofit





REST

Retrofit



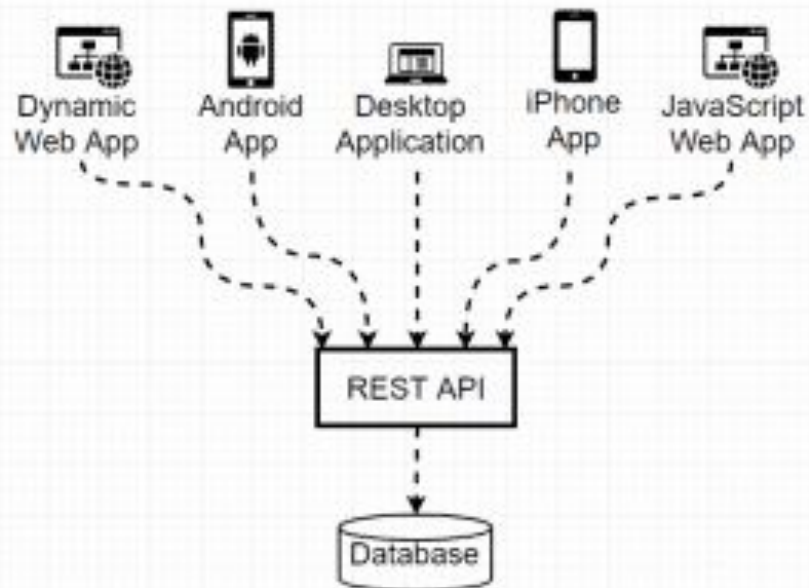
Sin estado





REST

Retrofit





REST

Retrofit



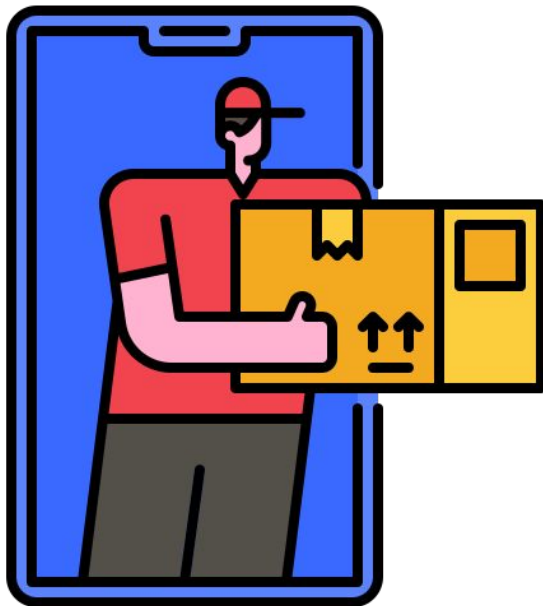
¿Qué es API Rest?





REST

Retrofit





REST API Methods



GET

Receive information
about an API resource



POST

Create an
API resource



PUT

Update an
API resource



DELETE

Delete an
API resource



REST

Retrofit





REST

Retrofit





HTTP response codes

HTTP code	Meaning
200	OK
4xx	Bad request (client's fault)
5xx	Failed request (server's fault)
401	Unauthorized request
404	Resource not found
500	Internal error (bug)
503	Server overloaded



REST

Retrofit



¿Qué es un Retrofit?





REST

Retrofit



Instalación





REST

Retrofit



implementation 'com.squareup.retrofit2:retrofit:2.9.0'
implementation 'com.squareup.retrofit2:converter-gson:2.9.0'





REST

Retrofit



Paso 1





REST

Retrofit



ApiService.kt

```
interface ApiService {  
    @POST("login")  
    fun login(@Body loginRequest: LoginRequest): Call<LoginResponse>  
}
```





REST

Retrofit



Paso 2





REST

Retrofit

```
ApiClient.kt

object ApiClient {
    private const val BASE_URL = "https://api.com/"

    private var retrofit: Retrofit? = null

    fun getClient(): ApiService {
        if (retrofit == null) {
            retrofit = Retrofit.Builder()
                .baseUrl(BASE_URL)
                .addConverterFactory(GsonConverterFactory.create())
                .build()
        }
        return retrofit!!.create(ApiService::class.java)
    }
}
```



REST

Retrofit



Paso 3





REST

Retrofit

```
MainActivity.kt

// Crear una instancia de ApiService
val apiService = ApiClient.getClient()

// Realizar la solicitud de inicio de sesión
val call = apiService.login("correo", "contraseña")
call.enqueue(object : Callback<LoginResponse> {
    override fun onResponse(call: Call<LoginResponse>, response: Response<LoginResponse>) {
        if (response.isSuccessful) {
            val loginResponse = response.body()
            // Procesar la respuesta del inicio de sesión
        } else {
            // Manejar el error de respuesta
        }
    }

    override fun onFailure(call: Call<LoginResponse>, t: Throwable) {
        // Manejar el error de red
    }
})
```



REST

Retrofit

Documentación

[Retrofit \(square.github.io\)](https://square.github.io/retrofit/)



Sesión 1



Fundamentos de Java

Sesión 3



Ciclo de vida del Activity

Sesión 5



Fragments

Sesión 2



Fundamentos de Android

Sesión 4



Componentes gráficos

Sesión 6



Servicios Web

Sesión 7

Sesión 8

Curso de Desarrollo Móvil en Android

