



QuickMatch

Language	Java, JavaScript y CSS
Framework	SpringBoot
Data Language	SQL / NoSQL
ORM	JPA (Java Persistence API)
IDE	IntelliJ
DATA IDE	DataGrip
Docker	Backend, Frontend y DB

Primera etapa

Elección de lenguaje, framework y ORM

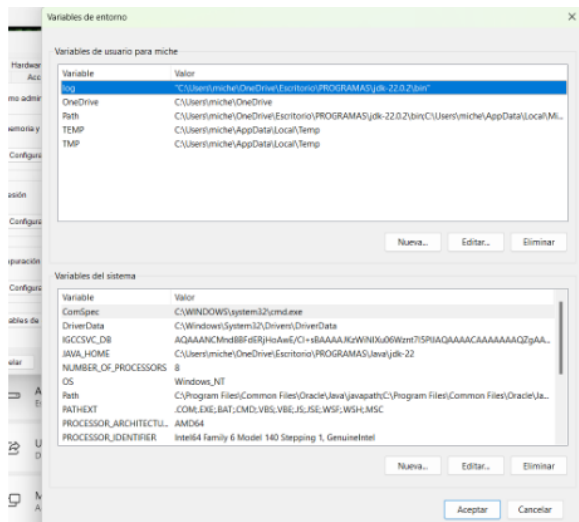
```

C:\Users\miche\OneDrive\Escritorio\DATOS2\QuickMart-Backend>docker run -p 8080:8080 quickmart

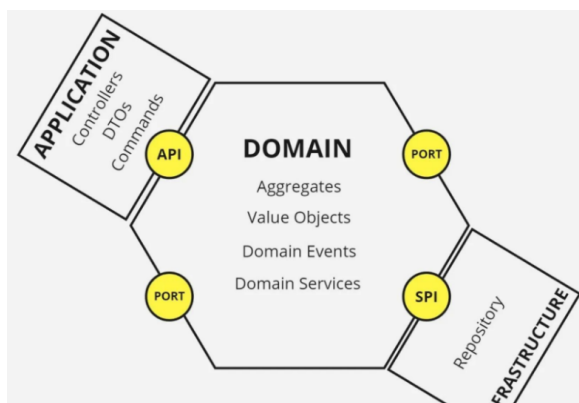
Spring
=====

:: Spring Boot ::      (v3.3.2)

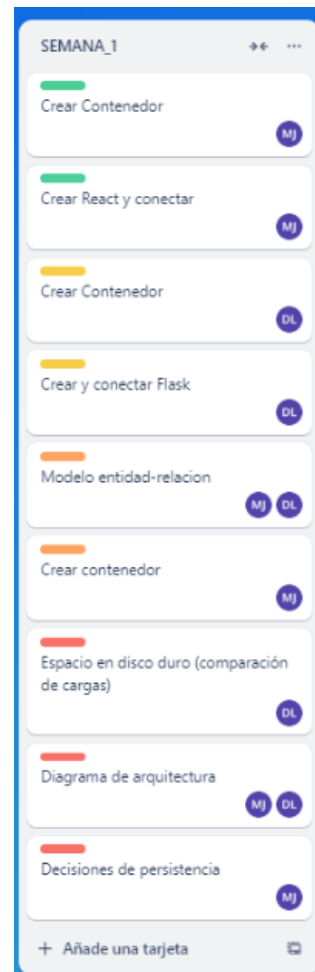
2024-08-16T16:22:05.131Z INFO 1 --- [QuickMart] [main] com.ufm.QuickMart.QuickMartApplication : Starting QuickMartApplication v0.0.1-SNAPSHOT using Java 17.0.2 with PID 1 (/app/QuickMart.jar sta
d by root in /app)
2024-08-16T16:22:05.139Z INFO 1 --- [QuickMart] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2024-08-16T16:22:06.406Z INFO 1 --- [QuickMart] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-08-16T16:22:06.625Z INFO 1 --- [QuickMart] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.26]
2024-08-16T16:22:06.686Z INFO 1 --- [QuickMart] [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2024-08-16T16:22:06.688Z INFO 1 --- [QuickMart] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1382 ms
2024-08-16T16:22:07.105Z INFO 1 --- [QuickMart] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path '/'
2024-08-16T16:22:07.134Z INFO 1 --- [QuickMart] [main] com.ufm.QuickMart.QuickMartApplication : Started QuickMartApplication in 2.697 seconds (process running for 3.597)
2024-08-16T16:29:43.942Z INFO 1 --- [QuickMart] [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2024-08-16T16:29:43.942Z INFO 1 --- [QuickMart] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2024-08-16T16:29:43.944Z INFO 1 --- [QuickMart] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 2 ms
  
```



Problemática con PATHS



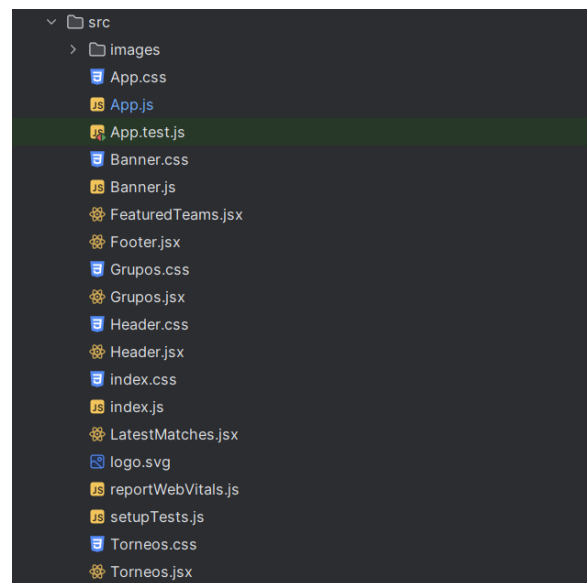
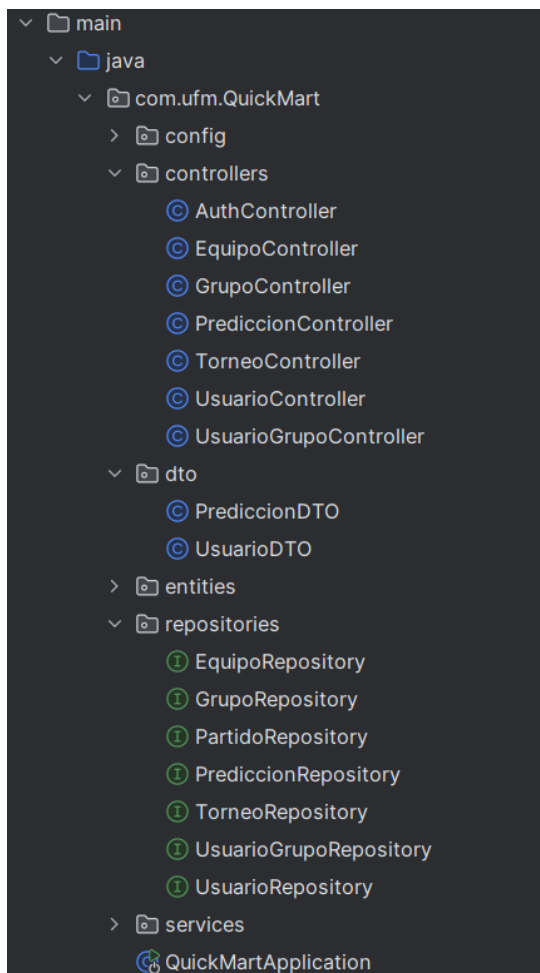
Enfoque en DDD



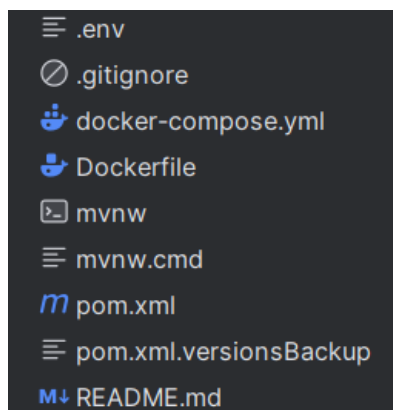
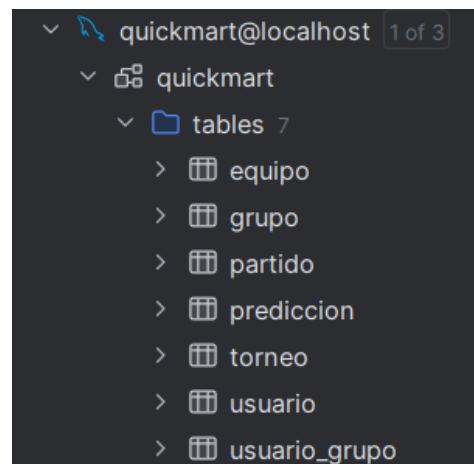
Organización en Trello

Estructura backend

- Config
- Controllers
- DTO
- Entities
- Repositories
- Services

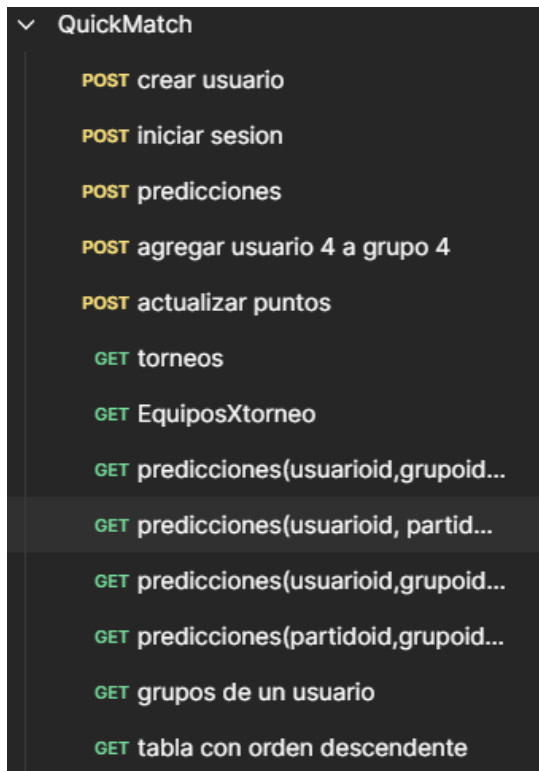


Estructura Frontend: Landing Page



Docker-compose/ Dockerfile para backend y
pom.xml para configuración

Pruebas ENDPOINTS (parte del API) > POSTMAN



Next Steps

Backend: → ~~Lunes 21 de octubre~~

- ~~Arreglar post: Actualizar Puntos (Error 403)~~
- ~~Agregar post: Crear grupo~~
- ~~Agregar Delete grupo~~

Frontend: → ~~Lunes 21 de octubre~~

- ~~Terminar Landing~~

ENTREGA 2:

- API Histo
- Frontend
- entender Indexación/Caché
- Aplicar indexación/caché
- Historial JMeter: con y sin caché
- Profiler (memoria): con y sin caché

	Redis	SIN CACHE	CON CACHE
	GET /predicciones		
Usuario, grupo, ronda, partido	http://localhost:8080/api/predicciones?usuarioId=1&grupold=3&ronda=Jornada2&partidold=10		
Usuario, partido	http://localhost:8080/api/predicciones?usuarioId=1&partidold=10		
Usuario, grupo, ronda	http://localhost:8080/api/predicciones?usuarioId=4&grupold=3&ronda=Jornada2		
partido, grupo, ronda	http://localhost:8080/api/predicciones?partidold=9&grupold=3&ronda=Jornada2		
	GET /equipos por torneos		
	http://localhost:8080/api/equipos/torneo/1		
	GET /grupos de un usuario		
	http://localhost:8080/api/usuarios/1/grupos		

PrediccionService.java

```

public List<Prediccion> getPredicciones(Long usuarioId, Long partidold) { 1usage 1 DanyHidalgo
    return prediccionRepository.findByUsuarioIdAndPartidold(usuarioId, partidold);
}

public List<Prediccion> getPrediccionesPorUsuarioYGrupo(Long usuarioId, Long grupold) { 1usage 1 DanyHidalgo
    return prediccionRepository.findByUsuarioIdAndGrupold(usuarioId, grupold);
}

public List<Prediccion> getPrediccionesPorPartidoYGrupo(Long partidold, Long grupold) { 1usage 1 DanyHidalgo
    return prediccionRepository.findByPartidoldAndGrupold(partidold, grupold);
}

public List<Prediccion> getPrediccionesPorUsuarioYGrupoYPartido(Long usuarioId, Long grupold, Long partidold) {
    return prediccionRepository.findByUsuarioIdAndGrupoldAndPartidold(usuarioId, grupold, partidold);
}

```

```

@Cacheable(value = "predicciones", key = "#usuarioId + '-' + #partidold") 1usage 1 DanyHidalgo
public List<Prediccion> getPredicciones(Long usuarioId, Long partidold) {
    return prediccionRepository.findByUsuarioIdAndPartidold(usuarioId, partidold);
}

@Cacheable(value = "predicciones", key = "#usuarioId + '-' + #grupold") 1usage 1 DanyHidalgo
public List<Prediccion> getPrediccionesPorUsuarioYGrupo(Long usuarioId, Long grupold) {
    return prediccionRepository.findByUsuarioIdAndGrupold(usuarioId, grupold);
}

@Cacheable(value = "predicciones", key = "#partidold + '-' + #grupold") 1usage 1 DanyHidalgo
public List<Prediccion> getPrediccionesPorPartidoYGrupo(Long partidold, Long grupold) {
    return prediccionRepository.findByPartidoldAndGrupold(partidold, grupold);
}

@Cacheable(value = "predicciones", key = "#usuarioId + '-' + #grupold + '-' + #partidold") 1usage 1 DanyHidalgo
public List<Prediccion> getPrediccionesPorUsuarioYGrupoYPartido(Long usuarioId, Long grupold, Long partidold) {
    return prediccionRepository.findByUsuarioIdAndGrupoldAndPartidold(usuarioId, grupold, partidold);
}

```

```

@CacheEvict(value = "predicciones", allEntries = true) 1usage 1 DanyHidalgo
public Prediccion hacerPrediccion(Long usuarioId, PrediccionDTO prediccionDTO) {

```

EquipoService.java

```

@Cacheable(value = "equipos", key = "#torneoId") 1usage 1 DanyHidalgo
public List<Equipo> obtenerEquiposPorTorneo(Long torneoId) {
    return equipoRepository.findByTorneoId(torneoId);
}

```

```

# Configuración de Redis para la caché (1 día)
spring.cache.redis.time-to-live=86400

```

```

# Configuración de Redis para la caché (1/2 día)
spring.cache.redis.time-to-live=43200

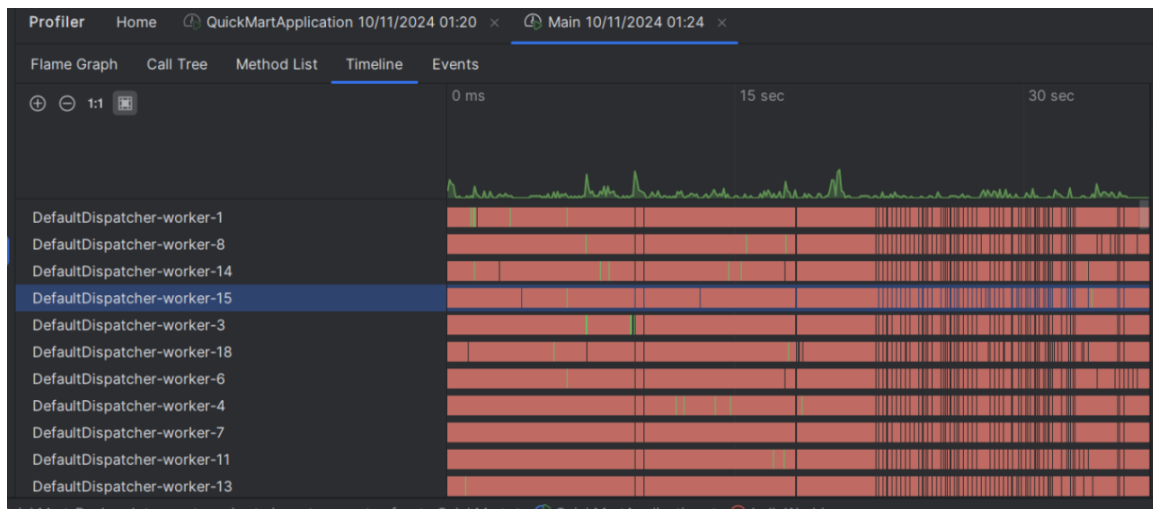
```

UsuarioService.java

```
// Método actualizado para obtener los grupos del usuario
@Cacheable(value = "gruposUsuario", key = "#usuarioId") 1 usage 1 DanyHidalgo
public Set<Grupo> obtenerGruposDelUsuario(Long usuarioId) {
```

Primera llamada a Profiler →

Haciendo llamadas a <http://localhost:8080/api/equipos/torneo/1> con 102 torneos 5 equipos cada uno (510 filas)



Sin ninguna llamada



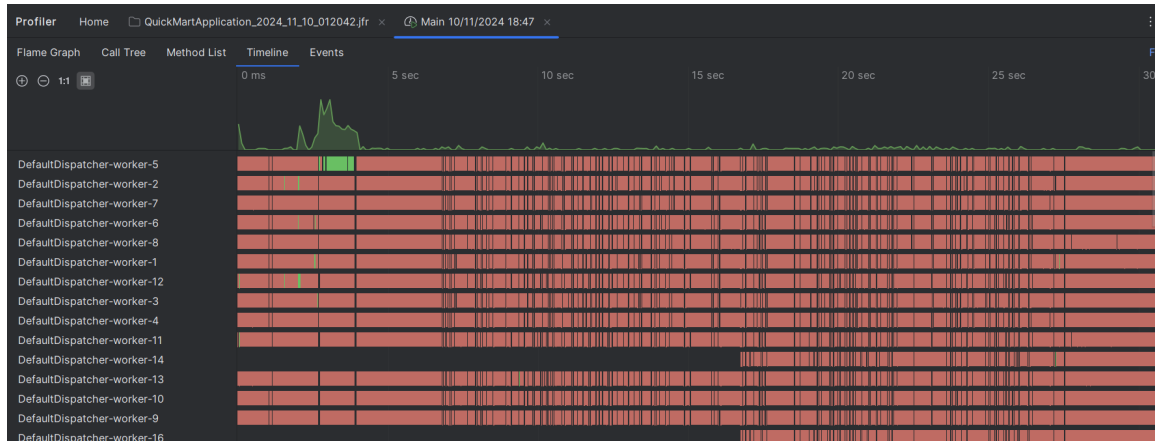
JMeter primera llamada a <http://localhost:8080/api/equipos/torneo/42>

```
Nombre del hilo:EquiposxTorneo 1-1
Comienzo de muestra:2024-11-10 15:57:44 CST
Tiempo de carga:6151
Connect Time:32
Latencia:6141
Tamaño en bytes:789
Sent bytes:197
Headers size in bytes:421
Body size in bytes:368
Conteo de muestra:1
Conteo de error:0
Data type ("text"|"bin"|""):text
Código de respuesta:200
Mensaje de respuesta:

HTTPSampleResult campos:
ContentType: application/json
DataEncoding: null
```

SEGUNDA LLAMADA

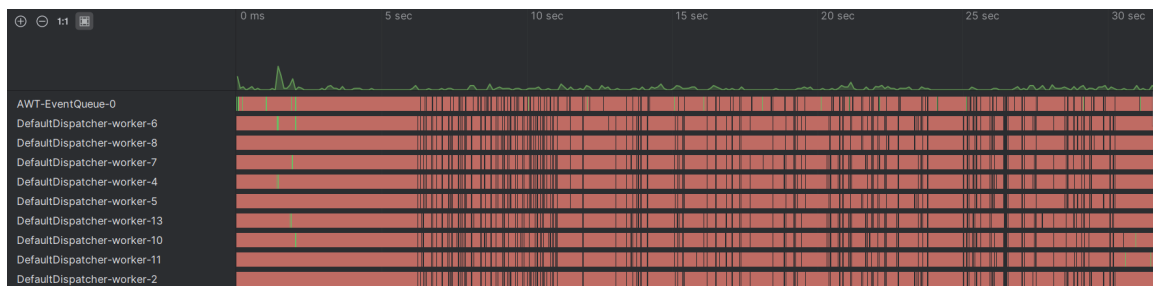
Haciendo llamadas a <http://localhost:8080/api/equipos/torneo/1> con 1101 torneos 5 equipos cada uno (5505 filas)



	Resultado del Muestreador	Petición	Datos de Respuesta
Nombre del hilo:EquiposxTorneo 1-1 Comienzo de muestra:2024-11-10 18:49:25 CST Tiempo de carga:76 Connect Time:5 Latencia:74 Tamaño en bytes:1199 Sent bytes:199 Headers size in bytes:421 Body size in bytes:778 Conteo de muestra:1 Conteo de error:0 Data type ("text" "bin" ""):text Código de respuesta:200 Mensaje de respuesta: HTTPSampleResult campos: ContentType: application/json DataEncoding: null	Nombre del hilo:EquiposxTorneo 1-1 Comienzo de muestra:2024-11-10 18:51:42 CST Tiempo de carga:66 Connect Time:2 Latencia:65 Tamaño en bytes:1199 Sent bytes:199 Headers size in bytes:421 Body size in bytes:778 Conteo de muestra:1 Conteo de error:0 Data type ("text" "bin" ""):text Código de respuesta:200 Mensaje de respuesta: HTTPSampleResult campos: ContentType: application/json DataEncoding: null		

USO DE @CACHEÉ

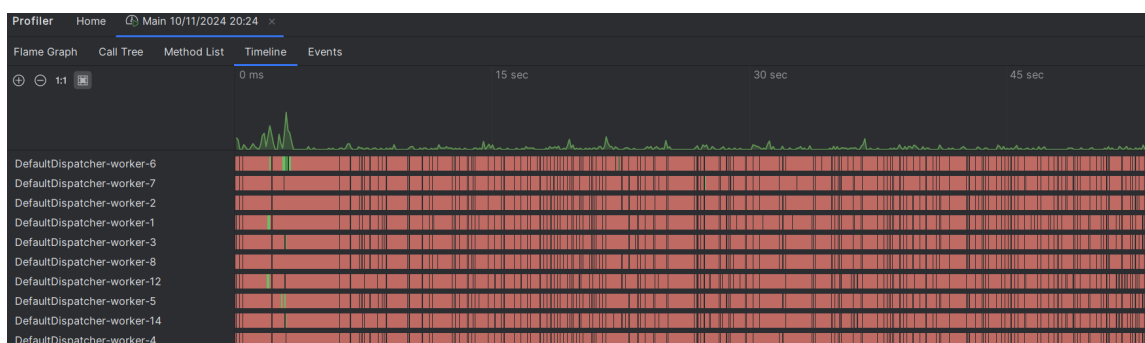
EquiposxTorneo



Nombre del hilo:EquiposxTorneo 1-1	Nombre del hilo:EquiposxTorneo 1-1
Comienzo de muestra:2024-11-10 19:26:19 CST	Comienzo de muestra:2024-11-10 19:26:27 CST
Tiempo de carga:40	Tiempo de carga:27
Connect Time:2	Connect Time:1
Latencia:39	Latencia:27
Tamaño en bytes:1199	Tamaño en bytes:1119
Sent bytes:199	Sent bytes:196
Headers size in bytes:421	Headers size in bytes:421
Body size in bytes:778	Body size in bytes:698
Conteo de muestra:1	Conteo de muestra:1
Conteo de error:0	Conteo de error:0
Data type ("text" "bin" ""):text	Data type ("text" "bin" ""):text
Código de respuesta:200	Código de respuesta:200
Mensaje de respuesta:	Mensaje de respuesta:
HTTPSampleResult campos:	HTTPSampleResult campos:
ContentType: application/json	ContentType: application/json
DataEncoding: null	DataEncoding: null

PREDICCIONES

Sin @Caché



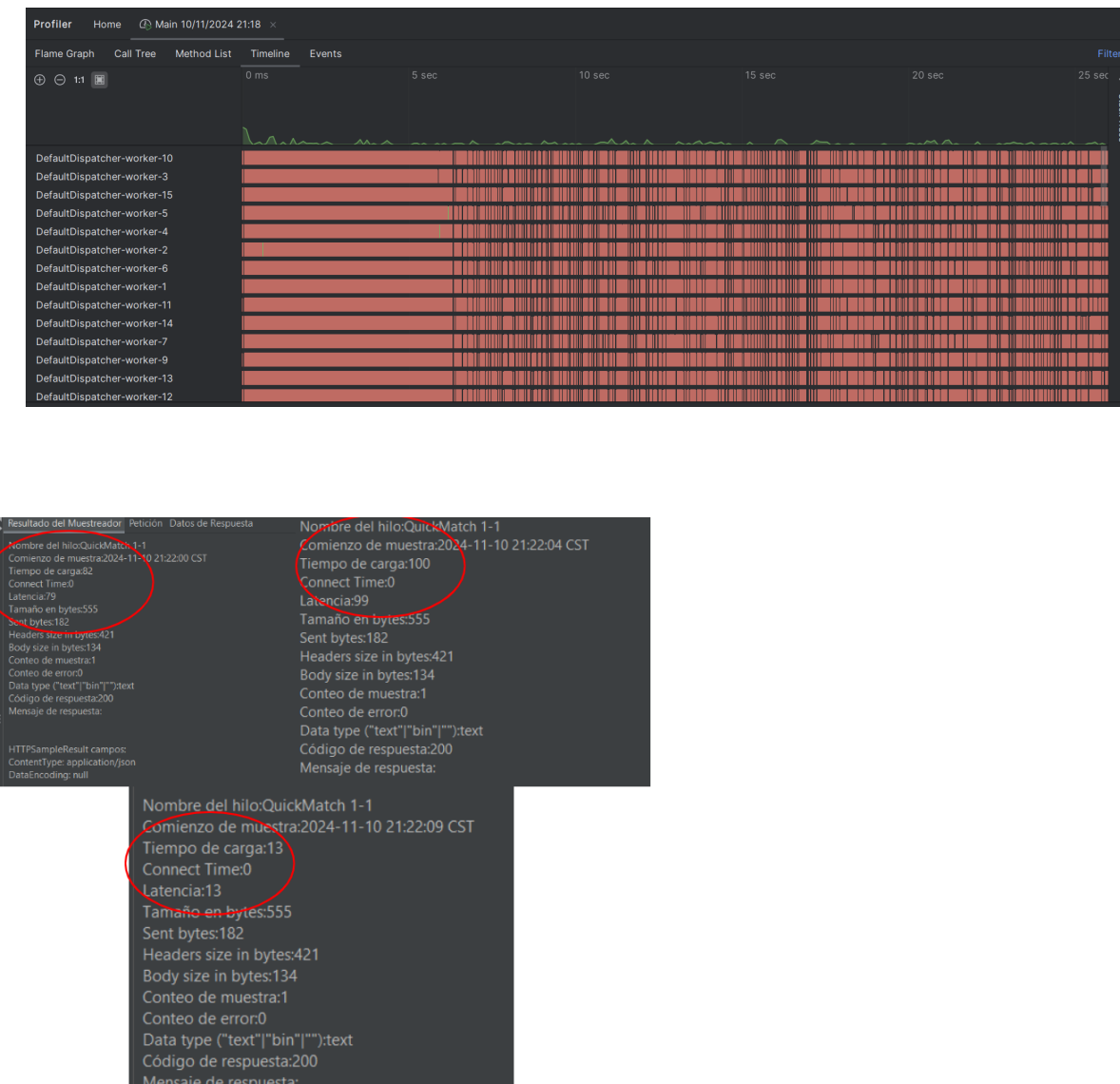
```

Nombre del hilo:QuickMatch 1-1
Comienzo de muestra:2024-11-10 21:12:15 CST
Tiempo de carga:107
Connect Time:0
Latencia:104
Tamaño en bytes:555
Sent bytes:182
Headers size in bytes:421
Body size in bytes:134
Conteo de muestra:1
Conteo de error:0
Data type ("text"|"bin"|""):text
Código de respuesta:200
Mensaje de respuesta:

```

PREDICCIONES

Con @Caché

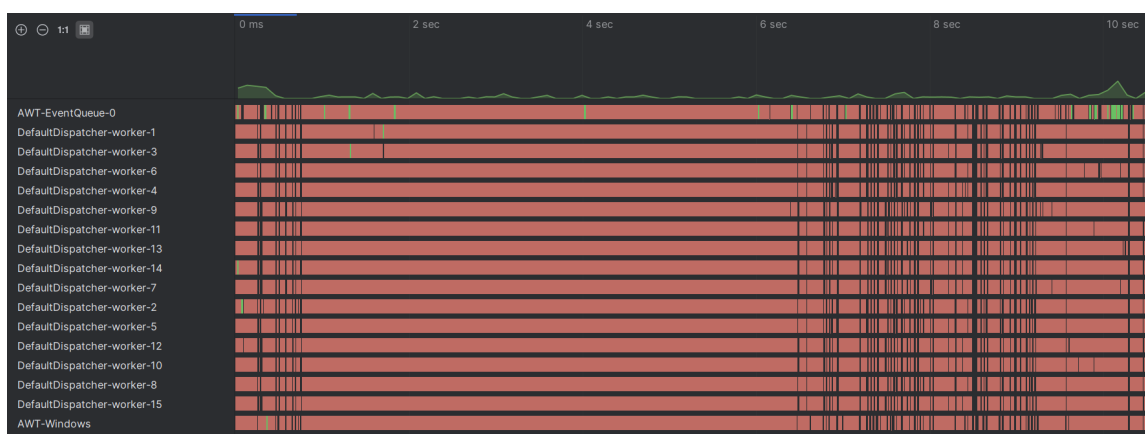


GRUPOS POR USUARIO SIN @CACHE



Nombre del hilo:QuickMatch 1-1	Nombre del hilo:QuickMatch 1-1
Comienzo de muestra:2024-11-10 23:34:50 CST	Comienzo de muestra:2024-11-10 23:35:11 CST
Tiempo de carga:202	Tiempo de carga:192
Connect Time:0	Connect Time:0
Latencia:201	Latencia:191
Tamaño en bytes:1222	Tamaño en bytes:1222
Sent bytes:137	Sent bytes:137
Headers size in bytes:421	Headers size in bytes:421
Body size in bytes:801	Body size in bytes:801
Conteo de muestra:1	Conteo de muestra:1
Conteo de error:0	Conteo de error:0
Data type ("text" "bin" ""):text	Data type ("text" "bin" ""):text
Código de respuesta:200	Código de respuesta:200
Mensaje de respuesta:	Mensaje de respuesta:
HTTPSampleResult campos:	HTTPSampleResult campos:
ContentType: application/json	ContentType: application/json
DataEncoding: null	DataEncoding: null

GRUPOS POR USUARIO CON @CACHE



Nombre del hilo:QuickMatch 1-1	Nombre del hilo:QuickMatch 1-1
Comienzo de muestra:2024-11-10 23:44:18 CST	Comienzo de muestra:2024-11-10 23:44:50 CST
Tiempo de carga:212	Tiempo de carga:109
Connect Time:0	Connect Time:0
Latencia:209	Latencia:108
Tamaño en bytes:1222	Tamaño en bytes:1222
Sent bytes:137	Sent bytes:137
Headers size in bytes:421	Headers size in bytes:421
Body size in bytes:801	Body size in bytes:801
Conteo de muestra:1	Conteo de muestra:1
Conteo de error:0	Conteo de error:0
Data type ("text" "bin" ""):text	Data type ("text" "bin" ""):text
Código de respuesta:200	Código de respuesta:200
Mensaje de respuesta:	Mensaje de respuesta:
HTTPSampleResult campos:	HTTPSampleResult campos:
ContentType: application/json	ContentType: application/json
DataEncoding: null	DataEncoding: null