

# Métodos de Ordenação

Métodos de ordenação, são funções que tem como objetivo, organizar de forma ordenada, crescente ou decrescente, um conjunto de dados, um vetor, por exemplo.

Mostraremos aqui, a diferença de:

- Velocidade de execução;
- Comparações realizadas;
- Permutações realizadas;

Entre os seguintes 4 métodos de ordenação:

- Bubble Sort
- Merge Sort
- Quick Sort
- Heap Sort

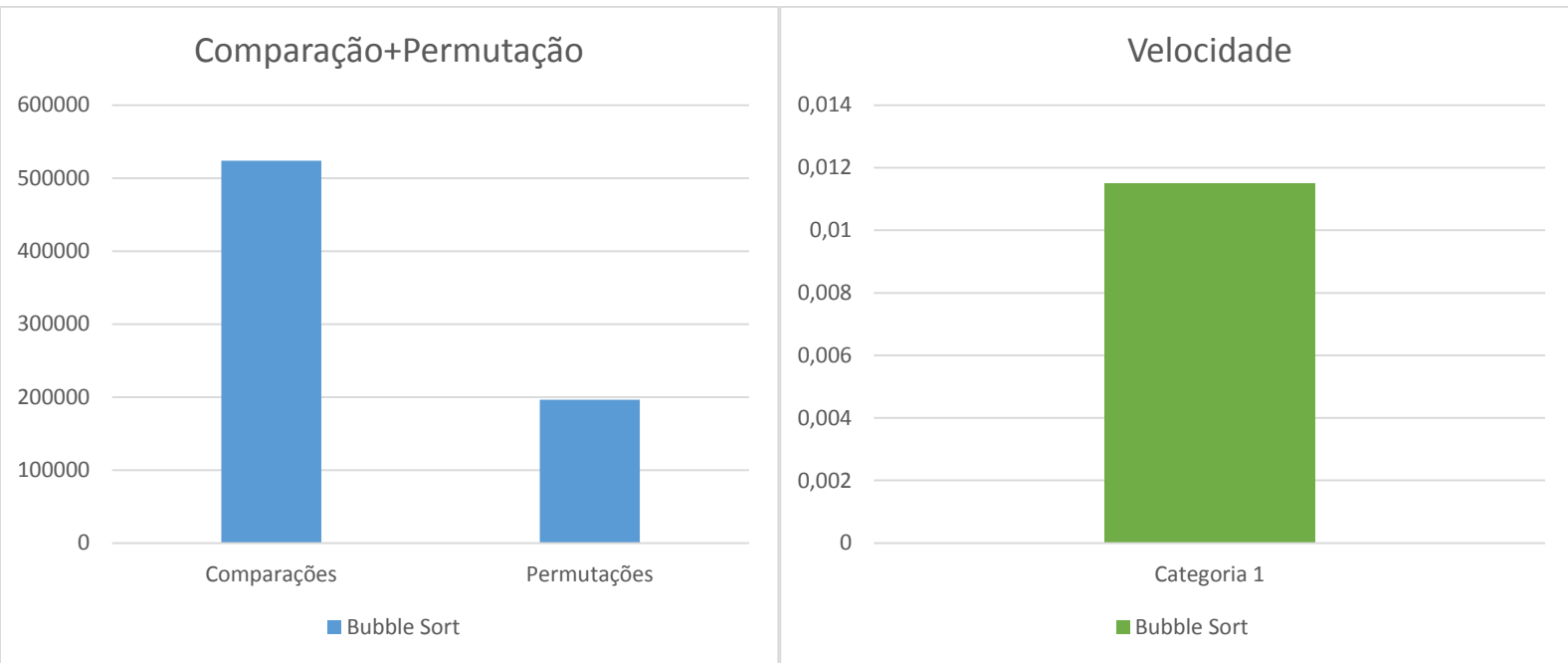
Utilizaremos, como exemplo, um vetor de 1024 posições mostrado abaixo.

Para cada método de ordenação, mostraremos dois gráficos, um contendo quantas comparações e permutações foram utilizadas para organizar completamente o vetor, e outra mostrando o tempo necessário para realizar tal ação.

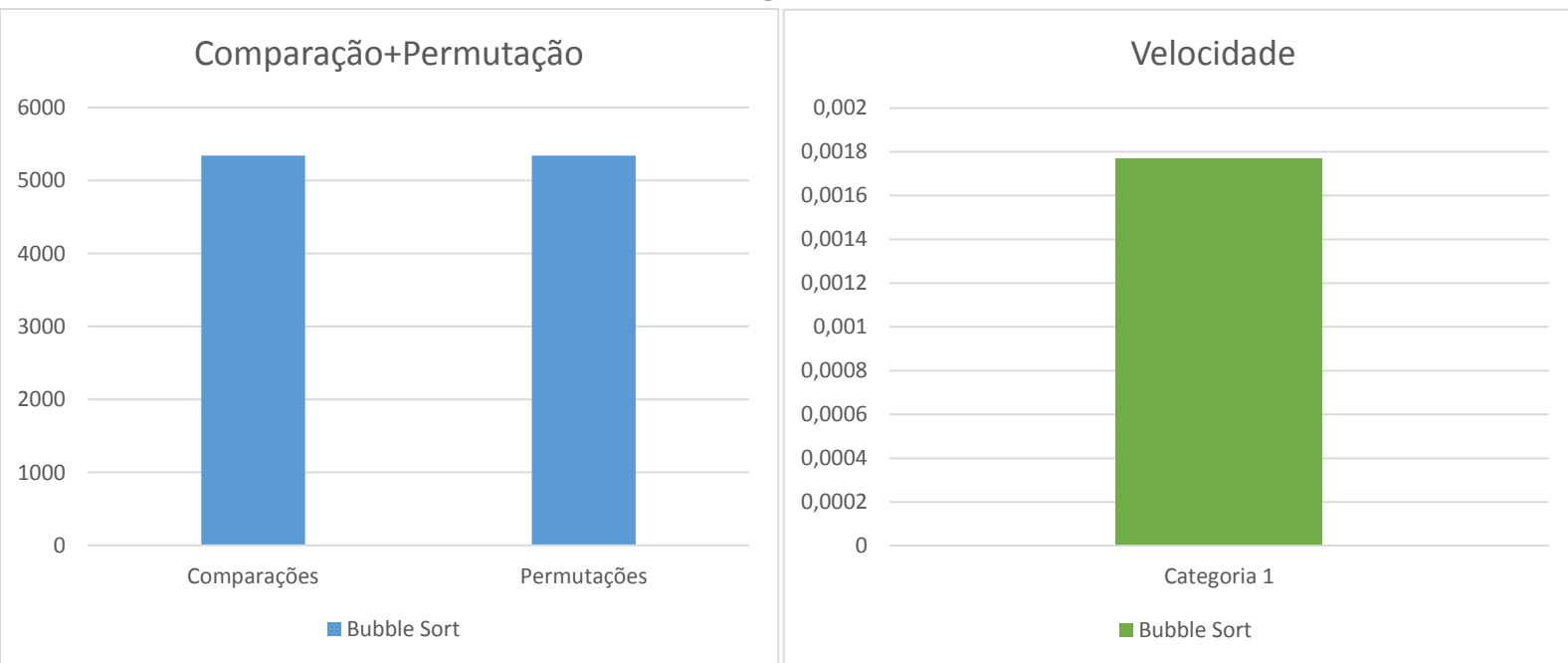
## Vetor Utilizado

868	609	99	665	873	747	12	578	72	387	15	441	82	843	86	467	876	811	544	933
289	907	562	798	67	787	298	680	110	330	248	277	289	442	202	462	593	13	730	758
292	585	358	829	679	126	211	976	573	89	999	862	860	257	866	398	613	606	944	356
888	769	590	514	450	278	726	755	380	727	400	731	529	331	911	410	196	611	854	407
603	156	701	953	324	80	428	857	754	175	570	27	12	44	578	587	434	618	771	405
573	173	419	428	130	860	943	636	323	476	217	892	266	361	999	234	467	677	220	914
685	531	109	762	980	949	962	237	422	961	366	999	279	178	773	743	692	459	450	280
625	566	978	666	279	331	24	135	966	580	966	167	162	581	221	12	477	697	945	274
865	863	403	206	392	331	564	343	74	945	667	234	194	589	972	815	254	582	838	513
944	881	878	795	478	537	293	310	323	883	555	495	679	160	86	365	322	284	771	488
24	804	488	909	954	493	131	454	529	54	275	230	223	319	161	906	940	419	962	475
522	265	188	481	757	497	346	657	394	433	534	766	683	111	66	302	309	209	251	195
590	552	563	946	22	363	386	896	921	640	383	696	509	333	824	50	87	373	884	638
845	798	145	829	567	694	448	656	918	406	65	162	577	983	11	384	649	401	988	129
704	659	434	585	666	756	644	719	895	440	120	982	773	38	145	937	910	390	564	664
355	184	791	935	101	336	447	421	347	221	285	394	605	536	957	569	804	266	330	813
812	614	361	80	463	150	424	318	494	882	307	97	983	907	179	990	510	973	675	228
313	547	312	325	245	36	180	332	30	763	0	222	287	196	52	774	838	354	900	873
939	46	384	328	29	978	192	531	273	148	514	890	336	974	890	551	35	90	940	759
350	162	409	472	686	749	828	477	864	182	432	804	720	175	432	280	531	979	834	298
993	184	969	890	658	611	101	860	820	54	425	518	174	516	841	430	100	694	838	616
504	422	256	780	713	536	23	30	155	168	509	583	307	46	979	313	652	710	943	624
139	28	319	204	219	821	869	323	718	897	446	563	33	837	776	541	515	741	280	429
802	462	334	814	650	760	61	915	178	651	907	471	532	729	169	478	369	505	87	461
919	740	440	140	729	800	925	276	361	42	280	210	716	424	871	51	722	933	955	203
584	381	565	102	932	552	156	943	685	548	893	550	276	530	215	244	274	715	601	35
808	620	819	209	557	380	174	403	8	758	737	797	618	209	497	183	334	709	139	188
264	873	65	372	40	336	872	233	178	41	666	383	547	909	908	325	393	300	950	240
47	832	770	502	460	123	275	727	207	501	939	512	145	189	153	50	71	408	286	551
443	592	490	676	574	388	696	66	948	941	387	799	411	816	883	434	73	151	567	267
642	603	416	816	331	396	312	620	606	600	390	569	490	314	823	575	958	114	115	941
418	979	401	524	489	73	87	140	383	377	930	325	780	843	22	119	634	34	242	125
520	852	132	980	555	388	453	946	198	957	488	618	302	985	428	148	29	744	707	368
9	940	846	987	271	457	856	2	705	706	284	323	82	22	939	340	466	826	534	539
947	112	92	692	638	589	864	513	506	272	163	603	586	325	513	647	890	666	139	620
694	785	608	433	903	500	920	980	603	327	591	692	855	711	451	112	468	105	290	24
644	421	810	529	535	536	278	791	710	377	186	290	811	907	276	907	563	153	215	904
754	681	141	99	841	487	696	729	846	74	229	710	296	99	789	205	284	601	953	965
33	434	26	408	990	284	219	373	251	762	971	461	226	378	619	858	587	483	776	339
830	27	509	445	953	262	168	652	310	148	359	767	991	473	719	177	321	160	961	355
74	329	993	702	265	943	525	511	1	975	446	908	17	330	102	739	843	443	910	425
855	548	626	95	187	267	116	673	897	546	914	281	981	692	253	760	945	123	759	68
207	159	336	629	419	561	564	898	307	541	28	729	523	654	736	475	137	251	806	421
306	527	834	859	351	599	371	375	392	148	45	855	307	344	31	773	816	661	603	832
996	649	206	519	920	342	108	870	110	966	33	713	528	419	542	517	900	380	662	326
232	844	675	699	614	781	650	878	460	746	952	47	299	82	775	16	89	943	482	875
101	142	729	568	79	386	872	251	665	561	858	314	838	304	102	99	854	482	427	508
659	387	310	285	515	204	236	631	298	839	558	984	747	409	353	662	962	151	598	721
778	889	45	206	398	804	904	933	758	788	215	772	8	974	408	236	32	51	1	9
158	48	460	872	203	66	789	81	407	744	803	939	785	121	505	438	68	876	143	780
451	510	690	801	745	69	924	66	199	747	681	762	907	167	480	646	1	814	627	738
337	463	835	373																

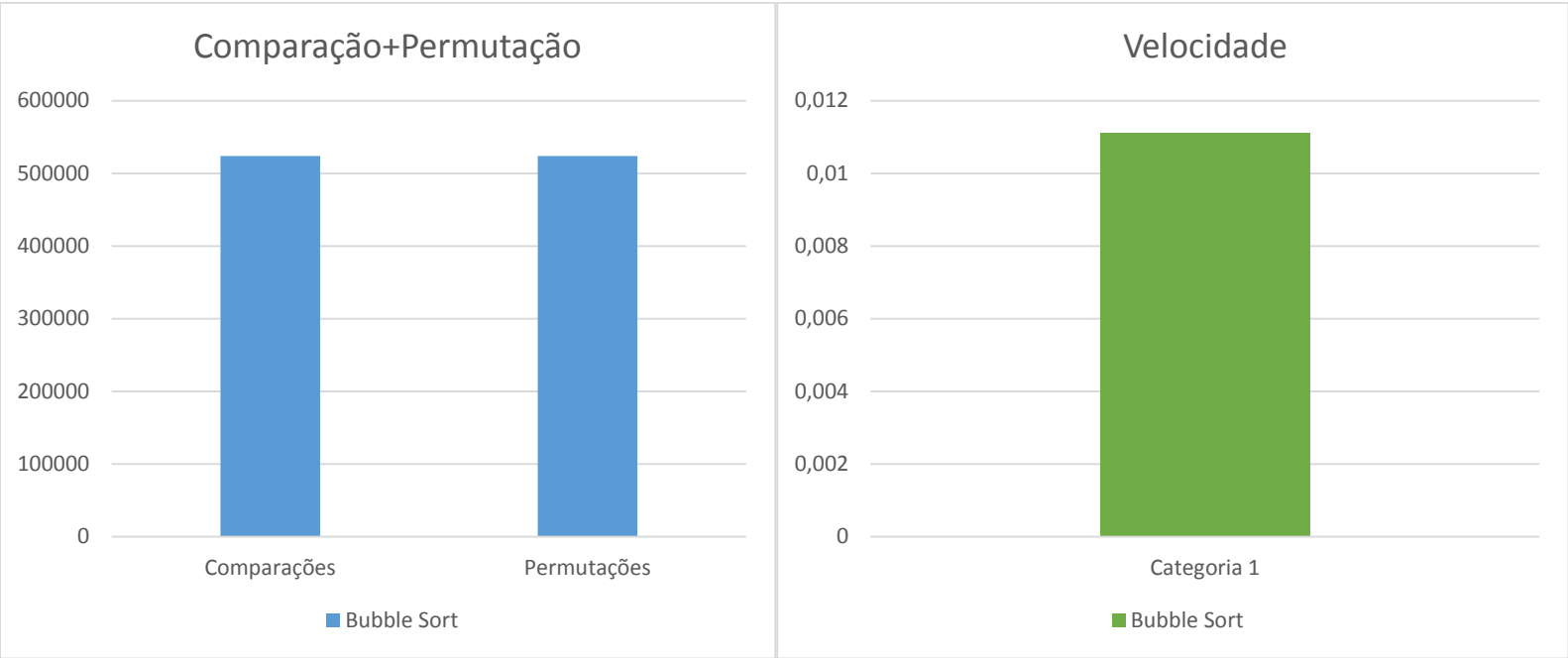
# Bubble Sort



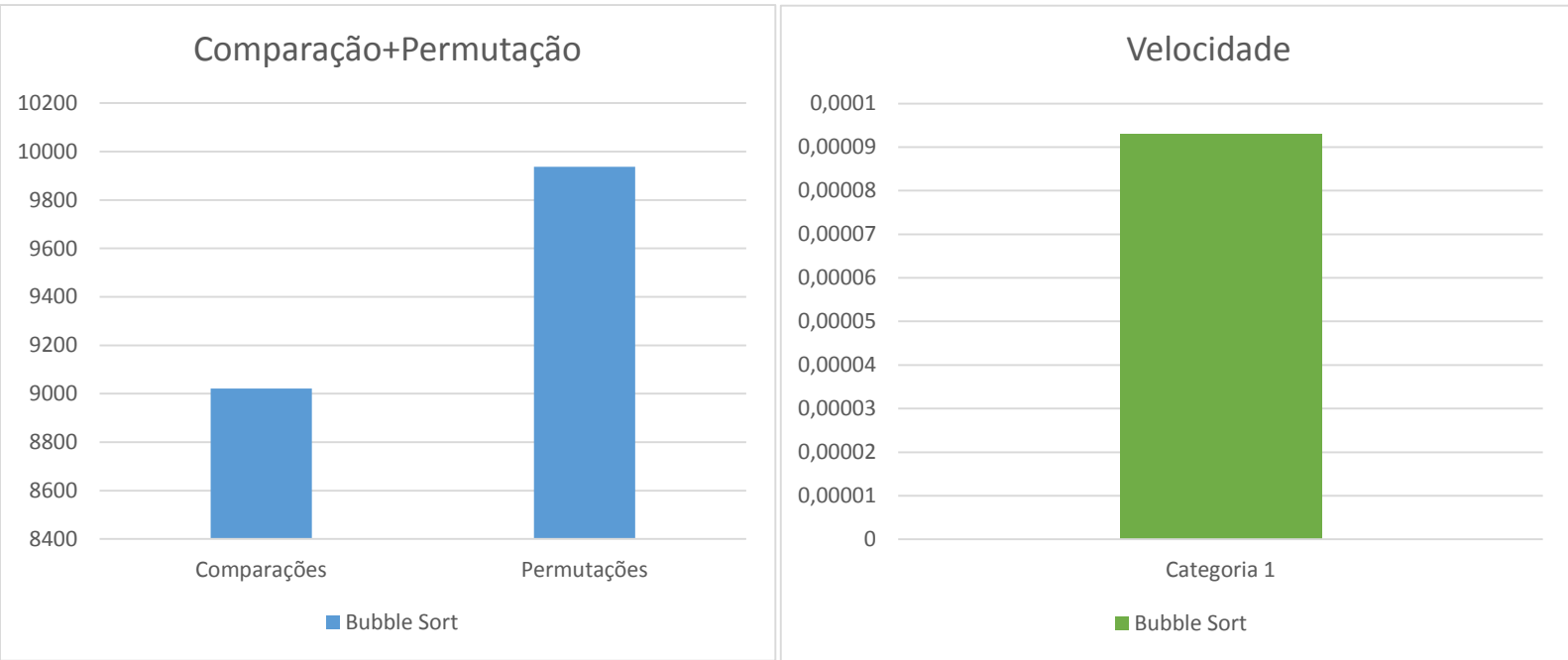
# Merge Sort



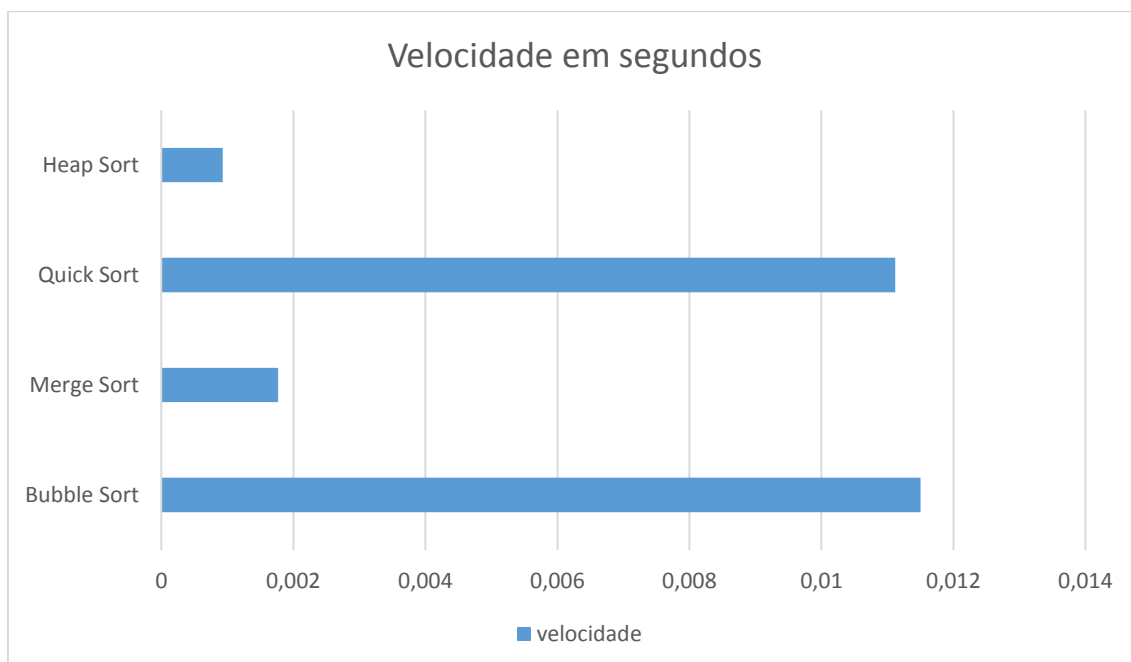
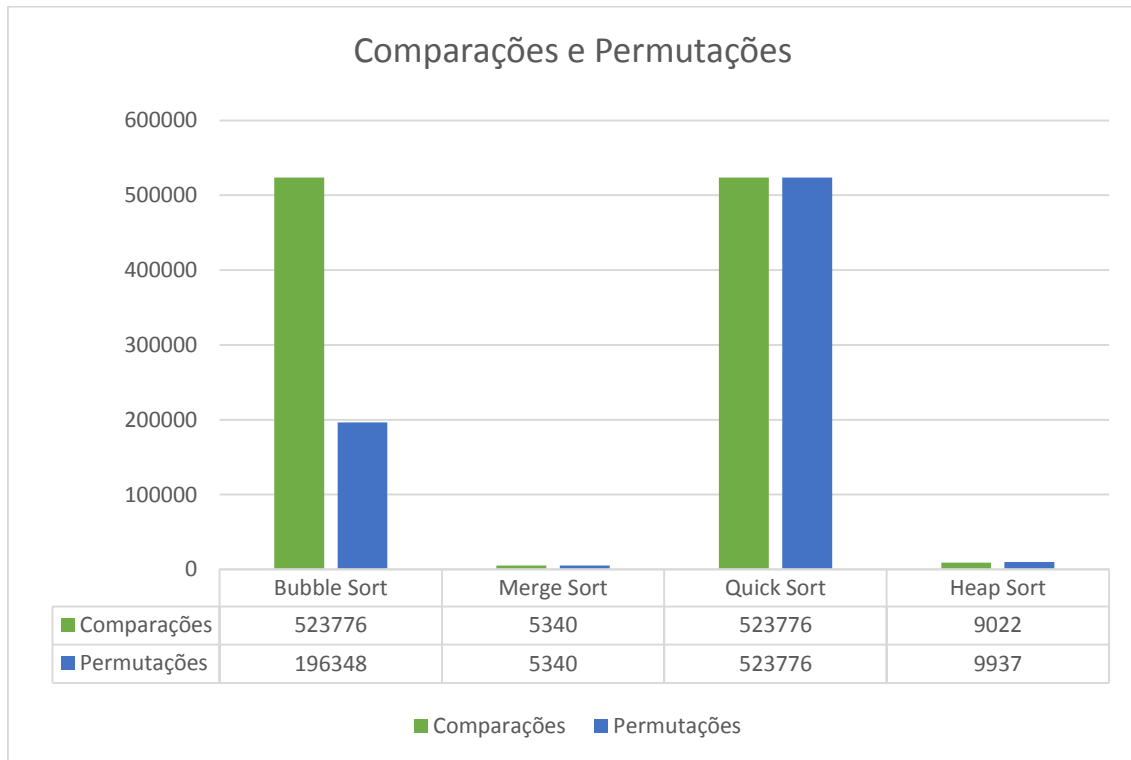
# Heap Sort



# Quick Sort



# Comparação entre todos os Sorts



Como vemos, o Heap sort, é o algoritmo mais rápido, porém, o que utiliza menos permutações e comparações, é o Merge.