Click Here

Boy de programa natal

Esta página cita fontes, mas que não cobrem todo o conteúdo. Ajude a inserir referências (Encontre fontes: Google (notícias • livros • acadêmico • imagens livres • WP refs) • ABW • CAPES). (Setembro de 2020) Esta página ou seção foi marcada para revisão devido a incoerências ou dados de confiabilidade duvidosa. Se tem algum conhecimento sobre o tema, por favor, verifique e melhore a coerência e o rigor deste artigo.Considere colocar uma explicação mais detalhada na discussão. Prostituição masculina num banho turco nova-iorquino cerca de 1884. A prostituição masculina é cada vez mais recorrente em diversas regiões do mundo, tanto em países desenvolvidos como a Espanha, onde por vezes os prostitutos, comumente chamados de gigolôs (termo que na América do Sul designa proxenetas), frequentemente vêm de países subdesenvolvidos ou em desenvolvimento, como o Brasil,[1] até países africanos, onde mulheres e homens europeus praticantes de turismo sexual se tornam seus principais clientes por serem mais baratos que os imigrantes ilegais que atuam na Europa.[2] A história da prostituição masculina remonta aos primórdios da prostituição. Mesmo a Grécia Antiga possuía também uma grande quantidade de πόρνοι / pórnoi, prostitutos. Uma parte deles trabalhava para uma clientela feminina, encontrando-se atestada a existência de gigolôs desde a Época Clássica. Na comédia Pluto, Aristófanes coloca em cena uma mulher de idade avançada que gastou todo o seu dinheiro num amante que agora a rejeita. Contudo, a maioria dos prostitutos trabalhava para uma clientela masculina. Os jovens prostitutos no Período Edo do Japão foram chamadas de kagema. Seus clientes eram principalmente homens adultos. O período durante o qual os adolescentes eram considerados desejáveis estendia-se entre a puberdade e o aparecimento da barba, constituindo a ausência de pêlos um elemento de erotismo entre os gregos. São mesmo conhecidos casos de homens que tinham como amantes homens mais jovens que se mantinham depilados. Da mesma maneira que acontecia com a versão feminina, a prostituição masculina não era para os gregos objecto de escândalo. Os bordéis de rapazes existiam não apenas nas zonas do Piréu, Keramaikos, no monte Licabeto, mas um pouco por toda a Atenas. Um dos mais célebres destes jovens prostitutos é sem dúvida Fédon de Élis. Feito escravo durante a tomada da sua cidade, o jovem trabalhou num bordel até que Sócrates o conheceu, tendo o filósofo comprado a sua liberdade. O jovem tornou-se seu discípulo, tendo o seu nome sido atribuído a dos diálogos de Platão, o Fédon que narra os instantes finais da vida de Sócrates. Os prostitutos masculinos encontravam-se também sujeitos ao pagamento de uma taxa. A existência de uma prostituição masculina em larga escala revela que os gostos pederásticos não estavam restritos a determinada classe social. Os cidadãos que não tinham tempo ou disponibilidade para seguir os rituais da pederastia (observar os jovens no ginásio, fazer a corte, oferecer presentes), poderiam recorrer aos prostitutos, que à semelhança das prostitutas encontravam-se protegidos pela lei contra as agressões físicas. Outra razão que explica o recurso à prostituição relaciona-se com os tabus sexuais: os gregos consideravam a prática do sexo oral como um acto degradante. Assim, numa relação pederástica o erastés (amante mais velho) não poderia pedir ao erómenos que praticasse este acto, reservado aos prostitutos. Apesar do exercício da prostituição ser legal, era mesmo assim uma prática vergonhosa, encontrando-se associado aos escravos ou aos estrangeiros. Em Atenas tinha para um cidadão consequências políticas, nomeadamente a perda dos direitos cívicos (atimía). Na obra Contra Timarco, Ésquines defende-se dos ataques de Timarco com a acusação de este ter praticado a prostituição durante a juventude, devendo por isso ser excluído dos seus direitos políticos, como o de apresentar queixa contra alguém. Não é raro que prostitutos homossexuais mal intencionados estejam ligados à prática de crimes como chantagem, extorsão, roubo, agressões verbais ou físicas, e, até mesmo, homicídio, particularmente em sociedades ou países onde a homossexualidade é criminalizada ou socialmente desprezada. Casos dramáticos como as intrigas que levaram Oscar Wilde à prisão, à sua ruína e morte prematura, ou o brutal assassinato de Pier Paolo Pasolini, são exemplo disso. Muitas das suas vítimas são levadas ao suicídio. Em Portugal, a esse tipo de prostitutos chantagistas, na gíria, é dado o nome de arrebentas.[3] Tal como no caso das mulheres, os preços cobrados pelos serviços variam consideravelmente. Ateneu refere-se a um rapaz que oferecia os seus serviços por um óbolo, mas o valor é considerado demasiado baixo. Estratão de Sardes, autor de epigramas, refere uma transação por cinco dracmas. Uma carta do Pseudo-Ésquines estima em 3000 dracmas o dinheiro ganho por um tal Melanopo, provavelmente durante toda a sua carreira. Passando-se para o caso da Roma Antiga, a moralidade romana já tinha mudado no século IV, quando Amiano Marcelino critica amargamente os costumes sexuais dos taifalos (em latim: Taifali, Taifalae ou Theifali), uma tribo bárbara que habitava entre os Cárpatos e o Mar Negro, de que fazia parte a pederastia ao estilo grego.[4] Em 342 os imperadores Constantino II (r. 317–340) e Constâncio II (r. 337–361) introduziram legislação que castigava a homossexualidade passiva, possivelmente com a castração. Estas leis que foram ampliadas em 390 por Teodósio I (r. 378–395), condenando à fogueira todos os homossexuais passivos que se prostituíam em bordéis. Em 438, a lei passou a abranger todos os homossexuais passivos, e em 533 Justiniano (r. 527–565) passou a castigar todos os atos homossexuais com a castração e a fogueira.[5] ↑ «70% dos homens que se prostituem na Espanha são brasileiros - 11/01/2008 - BBC». Folha de S.Paulo. 28 de dezembro de 2024. Consultado em 28 de dezembro de 2024 ↑ «Cinqüentonas aderem ao turismo sexual na África. Sem camisinha». O Globo ↑ Carlão, Bárbara (20 de fevereiro de 2020). «"Maria", "Pedro", "Paula": como os homossexuais resistiram à ditadura salazarista». Público. Consultado em 12 de janeiro de 2021 ↑ "Homosexuality. A history", Colin Spencer, 1996, Londres: Fourth Estate, id = 1-85702-447-8 ↑ "Gleich und anders", Robert Aldrich (ed.), 2007, Hamburgo: Murmann, id = 978-3-938017-81-4 Polícia desmantelou rede de prostituição masculina na Espanha 70% dos homens que se prostituem na Espanha são brasileiros PF descobre rede de prostituição masculina em São Paulo Obtida de " Share — copy and redistribute the material in any medium or format for any purpose, even commercially. Adapt — remix, transform, and build upon the material for any purpose, even commercially. The licensor cannot revoke these freedoms as long as you follow the license terms. Attribution — You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits. You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation . No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material. Sabrina Salerno (2010) BiografiaNaixement(it) Sabrina Debora Salerno 15 març 1968 (57 anys)Gènova (Itàlia) Altres nomsSabrina Alçada174 cm Pes56 kg Color dels ullsAvellana Color de cabellsCabells castanys ActivitatOcupacióactriu, actriu de teatre, actriu de cinema, presentadora de televisió, artista d'estudi, compositora, model, productora discogràfica, cantant, soubrette Activitat1986 - GènerePop, Italo disco, Hi-NRG, Eurobeat, pop dance i estil lliure VeuContralt InstrumentVeu Segell discogràficCasablanca RTI Music (en) Lloc websabrinasalerno.com Sabrina Debora Salerno (nascuda el 15 de març de 1968),[1] coneguda comercialment com Sabrina, és una cantant, compositora, productora discogràfica, model, actriu i presentadora de televisió italiana. Durant la seva carrera, ha venut més de 20 milions de discos a tot el món i ha aconseguit deu èxits internacionals, inclosos tres números u. Internacionalment, és més coneguda pel seu senzill "Boys (Summertime Love)" de 1987, que va encapçalar les llistes a França i Suïssa, i va assolir el tercer lloc al Regne Unit. La cançó va anar acompanyada d'un vídeo musical provocador, que va establir la imatge de Sabrina com un símbol sexual. Altres enregistraments populars inclouen "All of Me (Boy Oh Boy)", "My Chico", "Like a Yo-Yo", "Gringo", "Siamo donne" i una versió de "Call Me" (Blondie). La Sabrina va ser criada per la seva tia a Gènova abans de trasladar-se a viure amb els seus avis a Sanremo. Va anar a viure amb la seva mare quan tenia 15 anys. Va cantar al cor de l'església catòlica local i a l'escola va formar un grup de pop amb els seus amics.[2] Després de guanyar un concurs de bellesa a la seva regió natal, Ligúria,[3] Sabrina va començar a fer de model i el 1984 va fer el seu debut televisiu al programa italià Premiatissima a Canale 5. El 1986, el seu senzill debut "Sexy Girl" cantat en anglès, es va estrenar. Produït per Claudio Cecchetto, es va convertir en un èxit del Top 20 a la seva Itàlia natal i va ser un modest èxit internacional. A finals de 1987, va llançar el seu primer àlbum d'estudi, Sabrina, que va ser cantat completament en anglès. A més de "Sexy Girl", l'àlbum inclòia el seu èxit internacional "Boys (Summertime Love)" (número 1 a França i Suïssa, i Top 5 en més de deu països més;[4]) i "Hot Girl", un èxit del Top 20 en alguns països europeus.[5] "Boys" va guanyar popularitat pel seu famós vídeo, que inclòia escenes sexualment suggeridores. La cançó ha venut més d'1,5 milions de còpies fins ara a tot el món i segueix sent l'èxit de la marca registrada de Sabrina. El 1988, Sabrina va rebre el premi "Millor cantant europea" durant l'esdeveniment Festivalbar. També va gaudir d'un altre èxit d'estiu a nivell europeu amb el senzill "All of Me (Boy Oh Boy)", produït pel trio Stock-Aitken-Waterman.[6] Matt Aitken va eloglar l'actuació de Sabrina a l'estudi, dient que "com a cantant era bastant decent", però va assenyalar que la seva roba modesta durant la sessió "no va complir la promesa que s'esperava".[7] Aitken va expressar una certa decepció per l'actuació de la cançó en relació amb l'èxit enorme de "Boys", pensant que "potser el valor de l'impacte havia desaparegut, o potser no van rodar l'escena adequada a la piscina per al vídeo".[7] Més tard, aquell any, es va publicar el seu segon àlbum d'estudi, Super Sabrina, i va mantenir la seva imatge com a símbol sexual europeu gràcies als vídeos descarats que acompanyaven èxits com "My Chico" (el seu senzill més alt a Itàlia)[8] i "Like a Yo-Yo", produït per Giorgio Moroder. Aquest últim es va convertir en el tema musical d'Odiens, un programa de televisió italià molt popular en hora de màxima audiència, en el qual també va aparèixer Sabrina. Gràcies a l'èxit de la seva música i modelatge, Sabrina va actuar a molts països europeus, entre d'altres al Festival de Jazz de Montreux el 1988 i a l'Estadi Olímpic de Moscou, Rússia, el 1989, on 50.000 persones es van reunir durant tres dies per veure-la actuar. Fins i tot va protagonitzar un videojoc homònim per a ordinadors domèstics una mica arriscat per l'època.[9] El 1989, va protagonitzar la pel·lícula de comèdia italiana Fratelli d'Italia, dirigida per Neri Parenti, al costat d'actors com Christian De Sica, Jerry Calà i Massimo Boldi. També va llançar un nou senzill, "Gringo", amb un èxit moderat i el seu primer àlbum de remixs anomenat Super Remix. El 1990, Sabrina va ser l'amfitriona del programa de televisió setmanal Ricomincio da 2 amb Raffaella Carrà a Rai 2, i va llançar un nou senzill anomenat "Yeah Yeah", que només va ser un èxit modest. El 1991 va marcar un punt d'inflexió en la carrera de Sabrina, quan va gravar un duet amb la cantant italiana Jo Squillo, "Siamo donne", el seu primer llançament en italià. Van interpretar la cançó juntes al Festival de Música de Sanremo de 1991 amb molts èxits. El tercer àlbum d'estudi de Sabrina, Over the Pop, va ser llançat el mateix any, i per primera vegada se li va permetre coescriure i produir algunes de les cançons. El desig d'independència de Sabrina i el distanciament de la seva carrera glamurosa va provocar un conflicte amb la seva direcció. Com a resultat, la promoció de l'àlbum es va ressentir i tant l'àlbum com el senzill següent "Shadows of the Night" van ser fracassos comercials. Un nou senzill, "Cover Model", va ser llançat a França i Itàlia, però no va tenir èxit a les llistes i Sabrina es va separar del seu segell i direcció. Sabrina va llançar diversos senzills sense àlbum el 1994 i el 1995, inclòs el moderat èxit "Angel Boy" i una nova versió de "Boys". El 1996, va establir el seu propi estudi de gravació a Treviso amb el seu aleshores marit Enrico Monti, i va llançar el seu primer àlbum en italià, Maschio dove sei, que mostrava el seu so pop rock més madur. L'àlbum i els seus dos senzills, "Fatta e rifatta" i el tema principal, van ser sorprenentment aclamats per la crítica, però van ser fracassos comercials per falta de promoció i distribució. L'àlbum es tornaria a publicar l'any següent com a Numeri, amb una nova cançó del títol. Sabrina va debutar com a actriu de teatre a la comèdia I cavalieri della Tavola Rotonda, interpretant el paper de Morgan le Fay, i després va presentar el programa de jocs Il mercante in fiera. El 1998, Sabrina va participar en una comèdia anomenada Uomini sull'orlo di una crisi di nervi que també va tenir molt èxit, i posteriorment va presentar el programa de televisió Cocco di mamma a Rai 1. En aquest programa cantaria un estiu italià i internacional. èxits del passat, el que la va portar a llançar un retorn a la música. El 1999, Sabrina va llançar un nou àlbum, A Flower's Broken. Encara que el vídeo musical que acompanya la cançó "I Love You" va despertar cert interès, tant l'àlbum com el senzill no van tenir èxit, en gran part a causa de la dissolució del seu segell RTI Music. Sabrina va aparèixer al programa de televisió britànic Eurotrash en el qual va interpretar "I Love You". Actuació en octubre de 2010 (Moscou) A Wikimedia Commons hi ha contingut multimèdia relatiu a: Sabrina Salerno ↑ «Sabrina Salerno - Biography - IMDb». imdb.com. [Consulta: 21 febrer 2012]. ↑ Sunday World (Dublin) 17 July 1988 Page 18 ↑ «Il Lido di Genova: da stabilimento esclusivo a gigante decaduto» (en italià). Primocanale.it, 10-05-2013. Arxivat de l'original el 26-01-2023. [Consulta: 5 febrer 2023]. ↑ «Sabrina - Boys» (en francès). lescharts.com. [Consulta: 26 juliol 2015]. ↑ «Sabrina - Hot Girl» (en francès). lescharts.com. [Consulta: 26 juliol 2015]. ↑ «A Journey Through Stock Aitken Waterman: Ep 37: Love, Truth And Honesty to All Of Me on Apple Podcasts» (en anglès australià). Apple Podcasts. [Consulta: 30 agost 2022]. ↑ 7,0 7,1 «A Journey Through Stock Aitken Waterman: Ep 41: Looking Back with Matt Aitken on Apple Podcasts» (en anglès australià). Apple Podcasts. [Consulta: 24 octubre 2022]. ↑ «Hit Parade Italia - Indice per Interprete: S» (en italià). hitparadeitalia.it. [Consulta: 26 juliol 2015]. ↑ «Sabrina for ZX Spectrum (1988)» (en anglès). www.mobygames.com. [Consulta: 20 març 2021]. Obtingut de « Alpine.data('audio', () => ({ \/** * This method is used to get the on-demand playback url for a given url. * This handles the presence of the playback token. * Calling the ondemand endpoint guarantees that the consumption type of the listen is set to ondemand. * When a valid playback token is present, the ondemand endpoint will have a way to identify the listener. *\/ getOndemandOrDownloadUrl(url) { return SP.Utils.getOndemandOrDownloadUrl(url); }, \/** * This method is used to persist alpine store data in the local storage scoped to the user. *\/ persist(val, key) { const userId = null || 'guest'; return Alpine.$persist(val).as(SP.Utils.getLocalStorageKey(userId, key)); }, \/** * This method is used to fetch and store the variant of a given playback url. * To detrmine the variant, we need to make a HEAD request to the playback url because we need to follow * the redirect to the actual audio file. At this point if the url contains the variant, we can extract it *\/ fetchVariant(src) { return fetch(src, { method: 'HEAD' }).then((response) => { Alpine.store('audio').variant = SP.Utils.extractVariant(response.url); }).catch(() => { Alpine.store('audio').variant = null; }); }, init() { const $audioRef = this.$refs.audio; const getOndemandOrDownloadUrl = this.getOndemandOrDownloadUrl; Alpine.store('audio', { \/\/ Persisted on local storage episodeQueue: this.persist([], 'audio.episode_queue'), activeEpisode: this.persist(null, 'audio.active_episode'), playback_token: this.persist(null, 'audio.playback_token'), episodesProgress: this.persist({}, 'audio.episodes_progress'), episodesDuration: this.persist({}, 'audio.episodes_duration'), episodesCurrent: this.persist({}, 'audio.episodes_current'), episodesCompleted: this.persist({}, 'audio.episodes_completed'), \/\/ Not persisted on local storage playbackRate: 1, playing: false, loading: false, variant: null, \/** * PUBLIC METHODS *\/ play() { if (this.playing) { return; } if (this.activeEpisode) { this._updateAudioSrc(this.activeEpisode); this._updateMediaSession(this.activeEpisode); this._trackAmplitudeEvent('played', this.activeEpisode); this._trackBackendPlaybackEvent('play', this.activeEpisode, this .getCurrent()); this.changePlaybackRate(this.playbackRate); $audioRef.currentTime = this.getCurrent(); $audioRef.play(); return; } else if (this.episodeQueue.length === 0) { return; } else { this.activeEpisode = this.episodeQueue[0]; this._dequeue(this.activeEpisode); this._updateAudioSrc(this.activeEpisode); this._updateMediaSession(this.activeEpisode); this._trackAmplitudeEvent('played', this.activeEpisode); this._trackBackendPlaybackEvent('play', this.activeEpisode, this .getCurrent()); this.changePlaybackRate(this.playbackRate); \/\/ If the episode was completed we start from the beginning \/\/ and mark it as not completed, so that we can track the progress \/\/ locally and the user can resume from where they left off if (this.getCompleted()) { this.resetProgress(); } $audioRef.currentTime = this.getCurrent(); $audioRef.play(); } }, pause() { this._trackAmplitudeEvent('paused', this.activeEpisode); this._trackBackendPlaybackEvent('pause', this.activeEpisode, this.getCurrent()); $audioRef.pause(); }, plus30() { if (!this.activeEpisode) { return; } this.setCurrentTime(this.getCurrent() + 30); this._trackAmplitudeEvent('seek_forward', this.activeEpisode); this._trackBackendPlaybackEvent('seek', this.activeEpisode, this.getCurrent()); }, minus10() { if (!this.activeEpisode) { return; } this.setCurrentTime(this.getCurrent() - 10); this._trackAmplitudeEvent('seek_backward', this.activeEpisode); this._trackBackendPlaybackEvent('seek', this.activeEpisode, this.getCurrent()); }, seek(progress) { if (!this.activeEpisode) { return; } const duration = $audioRef.duration || this.activeEpisode.duration \/ 1000; const current = duration * progress; this.setCurrentTime(current); this._trackAmplitudeEvent('seek_at', this.activeEpisode); this._trackBackendPlaybackEvent('seek', this.activeEpisode, this.getCurrent()); this.play(); }, changePlaybackRate(rate) { this.playbackRate = rate; $audioRef.playbackRate = rate; }, next(automatic = false) { this._trackAmplitudeEvent('next', this.activeEpisode, { automatic }); if (automatic) { this.setProgress(0); this.setDuration(null); this.setCurrent(0); } this._stop(); this.play(); }, playNow(episode) { if (this._equals(this.activeEpisode, episode)) { this.play(); return; } this._stop(); this._enqueueHead(episode); this.play(); this._emitEpisodePlayEvent(); }, playPause(episode) { if (this._equals(this.activeEpisode, episode) && !this.loading) { if (this.playing) { this.pause(); } else { this.play(); } } else { this.playNow(episode); } }, isEpisodePlaying(episode) { return this._equals(this.activeEpisode, episode) && this.playing; }, shouldDisplayLoading(episode) { return this._equals(this.activeEpisode, episode) && this.loading; }, shouldDisplayPlay(episode) { return ((!this._equals(this.activeEpisode, episode)) || (this._equals(this.activeEpisode, episode) && !this.playing && !this .loading)); }, shouldDisplayPause(episode) { return this._equals(this.activeEpisode, episode) && this.playing && !this .loading); }, isEpisodeActiveOrQueued(episode) { return this._isEpisodeActive(episode) || this.episodeQueue.some((queuedEpisode) => { return this._equals(queuedEpisode, episode); }); }, countLoadedEpisodes() { return this.episodeQueue.length + (this.activeEpisode ? 1 : 0); }, addToQueue(episode) { if (this._equals(this.activeEpisode, episode)) { return; } this._dequeue(episode); this.episodeQueue = [...this.episodeQueue, episode]; if (this.episodeQueue.length === 1 && !this.activeEpisode) { this.activeEpisode = this.episodeQueue[0]; this._dequeue(this.activeEpisode); } this._trackAmplitudeEvent('queue_added', episode, { queue_length: this.countLoadedEpisodes() }); }, removeFromQueue(episode) { this._dequeue(episode); this._trackAmplitudeEvent('queue_removed', episode, { queue_length: this.countLoadedEpisodes() }); }, moveUpInQueue(episode) { const index = this.episodeQueue.findIndex((queuedEpisode) => { return this._equals(queuedEpisode, episode); }); if (index === 0) { return; } const prev = this.episodeQueue[index - 1]; this.episodeQueue[index - 1] = episode; this.episodeQueue[index] = prev; }, moveDownInQueue(episode) { const index = this.episodeQueue.findIndex((queuedEpisode) => { return this._equals(queuedEpisode, episode); }); if (index === this.episodeQueue.length - 1) { return; } const next = this.episodeQueue[index + 1]; this.episodeQueue[index + 1] = episode; this.episodeQueue[index] = next; }, setCurrentTime(time) { if (isNaN(time)) { return; } \/\/ Ensure the time is within the bounds of the audio time = Math.min(Math.max(0, time), $audioRef.duration || this.activeEpisode .duration \/ 1000); this.setCurrent(time); $audioRef.currentTime = time; }, getProgress() { if (!this.activeEpisode) { return 0; } return this.episodesProgress[this.activeEpisode.episode_id]; }, getEpisodeProgress(episode) { return this.episodesProgress[episode.episode_id] || 0; }, getEpisodeCurrent(episode) { return this.episodesCurrent[episode.episode_id] || 0; }, getEpisodeDuration(episode) { return this.episodesDuration[episode.episode_id] || episode.duration \/ 1000; }, getEpisodeCompleted(episode) { return this.episodesCompleted[episode.episode_id] || false; }, getEpisodeMinLeft(episode) { const current = this.getEpisodeCurrent(episode); const duration = this.getEpisodeDuration(episode); return SP.Utils.formatTimeLeft(duration - current, SP.Viewer.language); }, setProgress(progress) { if (!this.activeEpisode) { return; } this.episodesProgress[this.activeEpisode.episode_id] = progress; }, getDuration() { if (!this.activeEpisode) { return null; } return this.episodesDuration[this.activeEpisode.episode_id] || this .activeEpisode.duration \/ 1000; }, setDuration(duration) { if (!this.activeEpisode) { return; } this.episodesDuration[this.activeEpisode.episode_id] = duration; }, getCurrent() { if (!this.activeEpisode) { return 0; } return this.episodesCurrent[this.activeEpisode.episode_id] || 0; }, setCurrent(current) { if (!this.activeEpisode) { return; } this.episodesCurrent[this.activeEpisode.episode_id] = current; }, getCompleted() { if (!this.activeEpisode) { return false; } return this.episodesCompleted[this.activeEpisode.episode_id] || false; }, setCompleted(completed) { if (!this.activeEpisode) { return; } this.episodesCompleted[this.activeEpisode.episode_id] = completed; }, updateEpisode(episode) { \/\/ We update the active episode if (this._isEpisodeActive(episode)) { this.activeEpisode = episode; } \/\/ We update episode in the queue this.episodeQueue = this.episodeQueue.map((queuedEpisode) => { if (this._equals(queuedEpisode, episode)) { return episode; } return queuedEpisode; }); \/\/ We update the progress, duration and current time for the non active episodes if (!this._isEpisodeActive(episode) && episode.progress) { const newProgress = (episode.progress \/ episode.duration) * 100 const newDuration = episode.duration \/ 1000; const newCurrent = episode.progress \/ 1000; const newCompleted = !!episode.completed_at; this.episodesProgress[episode.episode_id] = newProgress; this.episodesDuration[episode.episode_id] = newDuration; this.episodesCurrent[episode.episode_id] = newCurrent; this.episodesCompleted[episode.episode_id] = newCompleted; } }, getEpisodePlayButtonState(episode) { if (this.shouldDisplayLoading(episode)) { return 'LOADING'; } if (this.shouldDisplayPause(episode)) { return 'PAUSE'; } if (this.shouldDisplayPlay(episode) && this.getEpisodeCompleted(episode) || !this.getEpisodeCurrent(episode)) { return 'PLAY'; } return 'RESUME'; }, displayEpisodeProgress(episode) { if (this.isEpisodePlaying(episode)) { return true; } if (!this.getEpisodeCompleted(episode) && this.getEpisodeCurrent(episode)) { return true; } return false; }, displayEpisodeDuration(episode) { if (this.isEpisodePlaying(episode)) { return false; } if (!this.getEpisodeCompleted(episode) && this.getEpisodeCurrent(episode)) { return false; } return true; }, resetProgress() { this.setProgress(0); this.setCompleted(false); }, \/** * PRIVATE METHODS *\/ \/** * The media session API is used to display the current episode information on the lock screen * of the device. The current implementations are pretty limited and buggy so conside this as * a best effort. *\/ _updateMediaSession(episode) { if ('mediaSession' in navigator) { const metadata = { title: episode.title, artist: episode.show.title, artwork: [{ src: SP.Utils.getNuvolariImageUrl(episode.image_original_url, 'square_limited_160',), sizes: '160x160', type: 'image\/jpeg', }, { src: SP.Utils.getNuvolariImageUrl(episode.image_original_url, 'square_limited_480',), sizes: '480x480', type: 'image\/jpeg', }, { src: SP.Utils.getNuvolariImageUrl(episode.image_original_url, 'square_limited_720',), sizes: '720x720', type: 'image\/jpeg', },], }; navigator.mediaSession.metadata = new MediaMetadata(metadata,); navigator.mediaSession.setActionHandler('seekbackward', () => { this.minus10(); }); navigator.mediaSession.setActionHandler('seekforward', () => { this.plus30(); }); navigator.mediaSession.setActionHandler('nexttrack', () => { this.next(); }); navigator.mediaSession.setActionHandler('play', () => { this.play(); }); navigator.mediaSession.setActionHandler('pause', () => { this.pause(); }); } }, _emitEpisodePlayEvent() { window.dispatchEvent(new CustomEvent('episode-play')); }, _updateAudioSrc: (episode) => { const newSrc = getOndemandOrDownloadUrl(episode?.playback_url); if ($audioRef.src !== newSrc) { $audioRef.src = newSrc; } }, _trackAmplitudeEvent(action, episode, data = {}) { if (episode) { SP.Amplitude.sendEvent(`player_${action}`, { podcast_title: episode.show?.title, podcast_id: episode.show?.show_id, episode_id: episode.episode_id, episode_title: episode.title, ...data, }); } }, _trackBackendPlaybackEvent(action, episode, current) { \/\/ If the user is not authenticated we do nothing }, _equals(episode, other) { if (episode === null || other === null) { return false; } return episode.episode_id === other.episode_id; }, _isEpisodeActive(episode) { return this._equals(this.activeEpisode, episode); }, _enqueueHead(episode) { this._dequeue(episode); this.episodeQueue = [episode, ...this.episodeQueue]; }, _dequeue(episode) { this.episodeQueue = this.episodeQueue.filter((queuedEpisode) => { return !this._equals(queuedEpisode, episode); }); }, _update() { this.playing = !$audioRef.paused; this.loading = $audioRef.readyState < 4; if ($audioRef.duration) { this.setProgress($audioRef.currentTime \/ $audioRef.duration * 100); this.setDuration($audioRef.duration); this.setCurrent($audioRef.currentTime); } }, _reset() { this.playing = false; this.loading = false; this.variant = null; }, _ended() { this.loading = false; this.setCompleted(true); this.next(true); }, _stop() { if (!this.activeEpisode) { return; } this._dequeue(this.activeEpisode); this.activeEpisode = null; $audioRef.currentTime = 0; $audioRef.src = ''; this._reset(); } }); const update = () => { Alpine.store('audio')._update(); }; const loadedmetadata = () => { this.fetchVariant($audioRef.src); }; const ended = () => { Alpine.store('audio')._ended(); }; $audioRef.addEventListener('play', update); $audioRef.addEventListener('pause', update); $audioRef.addEventListener('loadedmetadata', loadedmetadata); $audioRef.addEventListener('timeupdate', update); $audioRef.addEventListener('ended', ended); $audioRef.addEventListener('error', ended); } })); Alpine.data('audio_token', () => ({ playbackTokenInterval: null, playbackProgressInterval: null, \/** * This method is used to fetch a playback token from the backend and store it in the alpine store. * The playback token is stored in the alpine store so that it can be used to generate the on-demand * playback url. * * We need to call it withCredentials since the cookie is the only way for the app to identify the user. *\/ fetchAndStorePlaybackToken() { return fetch(`\/playback\/token`, { method: \"POST\", withCredentials: true, headers: { 'Content-Type': 'application\/json', 'Accept': 'application\/json', 'X-Csrf-Token': window.livewireScriptConfig.csrf }, }).then((r) => { if (r.status !== 200) { return null; } return r.json(); }).then((r) => { Alpine.store('audio').playback_token = r || null; }).catch((e) => { Alpine.store('audio').playback_token = null; }) }, \/** * This method runs once when the page is loaded, so here we're calling * ensurePlaybackToken to fetch a playback token if there isn't any. * The playback token TTL is 6 hours, so we set an interval to check the * expiration of the playback token every 30 minutes. The token will be * refreshed if the token is to be expired in 1 hour or less, so this way * the user can't be without a token. *\/ ensurePlaybackToken() { const ONE_HOUR = 60 * 60 * 1000; const currentToken = Alpine.store('audio').playback_token; if (currentToken) { const expirationDate = new Date(`${currentToken.expires_at} UTC`); const now = new Date(); const diff = (expirationDate - now); if (diff > ONE_HOUR) { return; } } return this.fetchAndStorePlaybackToken(); }, init() { const FIFTEEN_MINUTES = 15 * 60 * 1000; const THIRTY_SECONDS = 30 * 1000; this.ensurePlaybackToken(); this.playbackTokenInterval = setInterval(() => { this.ensurePlaybackToken(); }, FIFTEEN_MINUTES); this.playbackProgressInterval = setInterval(() => { this._trackBackendPlaybackProgress(); }, THIRTY_SECONDS); Alpine.store('audio')._trackBackendPlaybackEvent = this._trackBackendPlaybackEvent; }, destroy() { clearInterval(this.playbackTokenInterval); clearInterval(this.playbackProgressInterval); }, \/** * PRIVATE METHODS *\/ _trackBackendPlaybackEvent(action, episode, current) { if (!episode || isNaN(current)) { return; } if (current < 0) { return; } fetch(`\/playback\/${episode.episode_id}\/events`, { method: \"POST\", withCredentials: true, headers: { 'Content-Type': 'application\/json', 'Accept': 'application\/json', 'X-Csrf-Token': window.livewireScriptConfig.csrf }, body: JSON.stringify({ event_type: action, progress: Math.floor(current * 1000), }) }).catch(() => {}) }, _trackBackendPlaybackProgress() { let audioStore = Alpine.store('audio'); if (!audioStore.playing) { return; } this._trackBackendPlaybackEvent('progress', audioStore.activeEpisode, audioStore .getCurrent()); } })); "}} wire:id=EZNBrbFCT0PJzJjtg9B1 x-data=audio> { SP.Utils.setDocumentShouldScroll(!opened); })> Sign up to keep listening, access your Library to pick up episodes right where you left off, and connect with your favorite creators. Experience the ultimate podcast listening on Spreaker! Info Notícias O ministro da Saúde, Ricardo Barros, anuncia nesta quarta-feira (27), em Natal (RN), ações de reforço ao tratamento para os pacientes com câncer. As medidas irão qualificar o acesso da população a procedimentos oncológicos no Sistema Único Saúde (SUS). Data: 27 de dezembro (quarta-feira) Local: Hospital Universitário Onofre Lopes - Av. Nilo Peçanha, 620, Petrópolis, Natal/RN Horário: 16h Assessor em viagem: Camila Rabelo (61) 99211-2946 Assessoria de imprensa do Ministério da Saúde (61) 3315-3580 / 2351